Digitalisation of education using mobile devices to improve learning outcomes

Benjamin Trubert

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.6.2017

Thesis supervisor:

Assoc. Prof. Keijo Heljanko

Thesis advisor:

M.Sc. Yafeng Wang

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Benjamin Trubert

Title: Digitalisation of education using mobile devices to improve learning outcomes

Date: 30.6.2017 Language: English Number of pages: 7+58

Department of Computer Science

Professorship: Computer Science

Supervisor: Assoc. Prof. Keijo Heljanko

Advisor: M.Sc. Yafeng Wang

Digitalisation is the last revolution for our industry. Hence, it does not impact only this sector of activity but also all parts of our society, such as the education system. The digitalisation of education had started by using media content, and with the democratisation of personal computers inside classrooms. However, it impacts the learning process, changing methods, and by requiring new skills. This changing process follows the evolution of technologies; and nowadays, a majority of students and teachers own mobile devices. They are a common tool which is used continuously in our daily life. The aim of this thesis is to evaluate the potential of mobile devices to enhance the learning methods. Specifically, when using a huge number of devices, i.e., one per end-user, it becomes necessary to manage them to provide for each user the minimum requirement setup. Therefore, Mobile Device Management can be adapted to provide the initialization and maintenance of a whole fleet of devices. Thus, it becomes easier to manage installed applications and share course contents to all devices simultaneously. These devices can be either personal or shared by the school.

Keywords: mobile device management, mobile learning, digitalisation

Preface

I want to thank Assoc. Prof. Keijo Heljanko and Yafeng Wang for their good guidance. I would like also to thank the team of Cuppla for their support.

Otaniemi, 31.7.2017

Benjamin Trubert

Contents

Abbreviations

ASM	Apple School Manager
BYOD	Bring Your Own Device
DEP	Device Enrolment Program
MDM	Mobile Device Management
ML	Mobile Learning
MOOC	Massive Open Online Course
OMA	Open Mobile Alliance
TEL	Technology Enhanced Learning
VPP	Volume Purchase Program

1 Introduction

1.1 Research Background

Learning is one of the major contributors to the development and evolution of societies. Learning methods always changed to fit the new requirements regarding knowledge and new technologies. This is why in the last twenty years, schools tend to acquire digital equipment, because of the democratisation of personal computer and later on of Internet access. However, nowadays, the new digitalization of societies is focused on mobile devices, and more recently on Internet of Things. One of the results of this digitalisation is the creation of new skills related to devices and their technologies. Thus, it directly impacts the learning process, changing either the support or the content.

Concerning the support, new ways of teaching and learning have been directly created thanks to this democratisation, such as online courses (e.g., MOOC [10]). In this case, digital tools are used to provide the content to a broader audience more universally. Digitalisation can also alter the kind of content used during classes. This Master's thesis will develop this approach, focusing on the utilisation of mobile devices.

Indeed, this digitalisation can profit of the high number of students who own mobile devices and so have constant access to Internet services[1]. As they are more familiar with the technology, the general idea is to use these mobile devices to adapt the learning process to new requirements of digitalisation. Therefore, the thesis will search the possible technologies used to integrate mobile devices into classes.

1.2 Research Methods

As this thesis focuses on the different technologies, the literature survey is the preferred method. It will be used to both define general properties of mobile learning as well as discuss existing solutions. It permits to develop on the different existing solutions, and highlight actual technical possibilities.

Of course, this research takes into account users needs, and the implementation corresponds to an existing market. However, these needs are considered here similarly compares to already developed solutions. Therefore, these were not updated with user surveys. As there are no real new requirements, the thesis will not focus on those and instead take the requirements which were addressed for already existing products.

The research goes from a wider point of view about mobile learning. It will show the theoretical definition as well as the main challenges of this topic. Then based on these challenges, it will focus on some technologies which are currently used for teaching. Problems which need to be resolved will be highlighted from these solutions. They will define the different criteria of the implementation later described.

[1] https://hotforsecurity.bitdefender.com/blog/children-with-both-smartphone-and-computer-spend-up-to-five-hours-a-day-on-devices-18247.html

1.3 Outlines

This thesis is divided into eight sections. Theoretical background of Mobile Learning is defined in Section 2. Section 3 highlights the main challenges related to mobile device learning. In Section 4, learning solutions designed for mobiles devices will be listed. Section 5 defines what mobile Device Management is and how it answers the different problems face with mobile learning. In Section 6, a technical implementation of mobile Device Management will be detailed. Section 7 compares and evaluates the implementation explained in Section 6 to the different competitors.

2 Background

The digitalisation of the learning process follows the technology and the spreading of devices. This digitalisation is firstly explained with Technology Enhanced Learning (TEL). This topic explains the integration of technology in the teaching methods.

Then, Mobile Learning (ML) focus on the utilisation of mobile devices. These devices are used to access the new digital content, as well as to provide some mobility to its user. Therefore, it can impact on the environment for studying.

Finally, the high utilisation of mobile devices can be solved by taking care of the Bring Your Own Device (BYOD) paradigm. BYOD policy allows every user may be able to bring and use their personal device as a learning mean.

2.1 Technology Enhanced Learning

TEL focuses on the integration of digital technologies in courses. This integration can be done by replacing the mean of teaching, as well as changing the process [19]. The first way tends to replace the traditional content, for example, adapting content from books to mobile devices. The second kind of integration enhances the process by bringing new features and way of teaching.

2.1.1 Digital Media

The major modification created by TEL is the kind of content that is used for courses. Indeed, contents, which are used or created for a course, are nowadays more and more based on digital media. Shared information can be as videos, sounds (e.g., courses previously recorded), websites, mobile applications and so on.

The apparition of this kind of content in schools was possible because of the democratisation of the usage of personal computers directly inside classrooms. However, by their size, cost and the maintenance, the democratisation of one machine per person is limited. Nowadays, mobile devices, such as smartphones and tablets, replaces the personal computers. These devices are easier to equip for schools and can be shared among students [15].

2.1.2 Digital Devices

TEL not only focuses on digital contents which can be developed but also on the usage of mobile devices. As explained in [7], one-to-one TEL is the key to obtain good results. This means each student needs to have his own device, and not a shared one with others students at the same time, to work in good conditions. This permits to develop more participation and collaborative works during classes. Indeed, peer-to-peer communications can be implemented to improve interaction between students.

It is possible to improve the device utilisation, in the case it belongs to a particular student. Therefore, he will be able to continuously have access to the content provided for the course, such as homework. This situation allows to fully use the capacity of mobile devices, by changing the way of teaching.

2.1.3 Access to Content

TEL mainly depends on the ways to access the content. It concerns first the device itself. The technology is always evolving, reducing some of the costs. Therefore, even if the prices of flagship increase, the average price for smartphones is decreasing[2], making mobile devices more accessible. This is why schools can develop programs to share devices to students at a lower cost. TEL also permits the democratisation of smartphones and tablets, so that a maximum number of students, as well as teachers, own a personal mobile device.

The access also concerns network connectivity. The Internet access can vary depending on the location[3], increasing inequalities. With mobile devices, most of the provided services tend to be web services which required a permanent online connection [5], making network access mandatory for a full utilisation of mobile devices. The implementation of mobile networks (such as 4G, 5G) facilitates the access to digital contents and thus make TEL ubiquitous. Therefore, it leads to and facilitates the development of ML.

2.2 Mobile Learning

ML focuses on the integration of mobile devices for TEL. However, it also adds the concept of mobility, changing from the traditional environment to outside classrooms. ML tends to update the teaching methods, bringing flexibility and increasing the possibilities.

2.2.1 Utilisation of Devices

The first meaning of ML is the mobility of the users and the mobility of the device used to access the content [11]. This content can be for example audio guides in museums, learning games for portable consoles, mobile applications [20]. In the case of schools, mobile devices are now the target to provide content and address new ways of teaching. Mobile devices promote communications among class, and so improve collaborative work and also feedback about lectures or assignments to teachers.

The development of ML is possible because of the democratisation of mobile devices owned by students [34]. More than 50% have an internet access by using a personal device. This number increased if we take into account school project which ensures every student to have a tablet or a computer access for learning purposes [28].

2.2.2 Environment

ML also affects the context where students can learn. The notion of formal/informal learning develops the idea that with ML, knowledge is no more concentrated in the classroom [17]. Formal learning corresponds to the traditional classroom where a

[2] https://www.statista.com/statistics/510668/smartphone-average-selling-price-worldwide/

[3] http://geography.oii.ox.ac.uk/?page=internet-population-and-penetration

teacher makes lectures to a class of students. There is little interactions between both subjects. Informal learning promotes the implication of students in what they are learning, by changing the environment and methods to acquire the knowledge, such as museum visits, i.e., learning by experience [30]. ML means that it is possible to have access to a course or content outside the school from a space point of view but also from a time point of view [31].

From a space context, ML by using mobile devices and ubiquitous internet access, allows users to be able to have access to content anywhere. It could be at home or in the library for assignment and group works, but also in museums where students can have access to more information, increasing interactions with what they are watching. From a time perspective, it allows students to have access to a course even after it finished and the possibility to continue his work after the end of working hours. It facilitates communications with his peers or teachers at any time, with messaging services.

Changing the environment can also mean adapting the class configuration, for example, students working together with smaller groups, mixing formal and informal learning [30]. Moreover, even if the vast majority of students and teachers possess and have access to computers, or even now mobile devices, there are still limitations. First limitations are those who are directly related to technology possibilities. The second kinds of limitations concern the users. Some of them can not have access to mobile devices or face more difficulties to understand and use digital solution for education.

An example of ML is Massive Online Open Classrooms (MOOC). This solution gets rid of any physical space and offer their content at any time and anywhere. It will be described in the Section 4.1.

2.3 Bring Your Own Device

BYOD is mainly used to ensure the one-to-one TEL property. In this case, the device belongs to its user, who can use it at school and his home. In case every student already owned a personal device, it increases the potentiality of integrating mobility.

2.3.1 Definition

BYOD policy was initially implemented in companies to avoid workers to keep constantly using multiple devices. Employees use their device to use the applications of the company and to have access to the infrastructure (e.g. data on a remote server) of the company [1]. The company can use a Mobile Device Management (MDM) solution to keep control on the data they share with their employees. This solution may be used by schools which are allowing students to use their personal device inside classrooms.

2.3.2 Advantages

BYOD brings several advantages either for enterprise or schools. The first advantage for institutions is the reduction of the investment price and maintenance costs. This

policy reduces the cost to equip everyone with a device. The majority of students already owned a personal device. Thus, it avoids a useless multiplication of devices for a similar user. Another advantage is the better performances measured for the productivity employees [23], while they are working with their own only one device. When they are using their personal devices, they are already familiar with the platform, and the global utilisation. Thus, there is no learning curve to adapt to a new product.

2.3.3 Drawbacks

However, BYOD has also some drawbacks; the main ones are about security and privacy issues [23]. These problems concern either one institution or its users. In the example of enterprise, the users need to trust the companies applications that they will not monitor personal data on the phone and download personal content. A company has to protect the access to their system and their critical data. They may face all kind of behaviour from their employees. The article [25] lists the major threats that exist when using an MDM solution to manage devices. The authors propose some security requirements which are needed to ensure the security and also the privacy of the system and its data.

As they are personal devices, they are not preconfigured to meet the requirement of utilisation of schools. Thus, if not managed, they may not be compatible with the minimum requirements to access the course content. Compatibility issues can also come from the diversity of devices, which can be different from one user to another. This difference can be from the platform (i.e., different OS, or version), as well as hardware specifications (e.g., sensors, cameras).

3 Challenges

This section presents the challenges which need to be solved to provide a solution for mobile learning using mobile devices. The described challenges will highlight the problem statement which will define the implementation of a possible solution.

3.1 Solution Requirements

The challenges concern either the technology requirement for the development as well as the user experience. As mobile learning concerns users, such as students and teachers, one key point is the ease of use of a solution. Concerning the technology requirements, mobile devices represent an ecosystem, so a mobile learning solution has to be able to integrate services as well as a cross-platform compatibility.

3.1.1 Adaptivity

The learning curve when testing a new solution is a major argument to take into account to know if end-users will accept to continue to use the solution after a test phase. The user experience is the centre point in the decision for someone to select a product for his daily use. Technical features, if too complicated to use, users will be reluctant to adopt them, so they will not be positive advantages compared to competitors. Does it bring better results in a shorter amount of time? What are the users (teachers and students) reactions regarding the usage of mobile devices in the classroom? Part of the teachers may be reluctant, adding a source of distraction, such as mobile devices with all their games and social networks, in their lessons. For education purpose, the ease of use of a solution is the most important point to address it to a vast majority of users. Indeed, it will concern different age sections and users from diverse backgrounds. On the contrary of other markets, it is harder to narrow the education market to a specific category of users.

3.1.2 Compatibility

The compatibility of a solution concerns its integration into a global ecosystem. Lots of tools already exist in the market; thus a right solution can provide an easy interaction with others tools users may use. For example, an application can not reinvent the wheel and compete with an already existing service from Google. Therefore, it is important to be able to use services that may be related and needed to offer a complete solution to the end-users. It also concerns the way to initiate, install/setup the new system and how to maintain it throughout the time of its runtime.

When talking about mobile devices learning, the first step is to have a compatible mobile device to use a solution. Depending on the project different requirements might be taken into consideration. In the case of BYOD policy, it may be important to be sure that a maximum of configurations of devices (hardware and software) are compatible.

3.2 Problem Statement

This part highlights the problems which need to be included in a solution of mobile learning. The problems addressed in the Master's thesis are divided into four categories.

User Management
The first one concerns users management. It will be described in the Section 6.1. The global system has to have only one user based. Moreover, to facilitate its utilisation, the system must have a separate role depending on the user type.

Device Management
The second category presented in the Section 6.2, concerns device management. To use mobile devices inside classrooms requires to be able to setup them, and ensure their maintenance. That is to say, the system has to register them, i.e., enrolling them. It also has to send control commands, to update them to a correct state (e.g., share content, manage data, lock a device).

Application Management
A crucial point when using mobile devices is the access to applications. The developed system has to incorporate an application management to ensure the installation of required apps. It can also be used to restrict the access to other applications, and so control the device usage. Application management process will be explained in the Section 6.3

Course Management
Finally, the last category concerns the course management in itself and will be discussed in Section 6.4. The content shared has to be compatible with the different existing platform, be universal. Moreover, to enhance this compatibility, it should be able to integrate others solutions to improve the content shared. This course management also has to include interaction, to benefit to mobile device possibilities of actions and communications.

4 Solutions

This section describes existing solutions of mobile learning. These solutions consist of contents focus on mobility, and contents designed for mobile devices.

The mobility is illustrated with the MOOC system, which is an Internet based solution. Then, examples of mobile application are presented. The last part shows solutions used by schools, which can be directly integrated into courses.

4.1 Massive Open Online Course

MOOC is opened to everyone around the world. That means that by definition, there is no geographical limitation, offering a complete mobility for the users. Learning is based on the environment and pace chosen by the users, which may be independent of each other.

4.1.1 Definition

MOOCs are courses which are provided online. They are based on web technology, i.e., digital courses. MOOC gained in popularity with the creation of courses from prestigious universities [6]. With this solution, the courses proposed to users are free and are open to everyone who wants to register, regardless the academic background, or the nationality.

These courses share many commons points with regular courses in schools. Some of the MOOC courses directly comes from already existing courses of universities. They most of the time consist of lectures with exercises, ended by a final exam. The main difference is that at the end if a user passes a course, he will receive a certification instead of validating credits for a diploma.

The content provided is, of course, digital content, and the user will consume regarding his own schedule. This paper [16] studies the kinds of videos which are the most appropriate to keep the attention of students. Thus, MOOC courses tend to answer the problem of which kind of content make more sense to be used on a digital platform. In addition to the content provided to students, MOOC platforms embed forum in order users to interact with each other, by asking questions or discussing problems.

One of the major interest of MOOC is the possibility to choose "personalised" content, and also how to get access to this content. Indeed, it is all based on user personal choice to attempt particular courses based on its motivations, allowing a certain independence. That means he will be free to study when he desires, only needed to respect some deadlines for some assignment or final exams. MOOC concept follows the digital trend of Internet where everything goes faster [22], similar to cloud philosophy (pay-as-you-go), learn-as-you-go.

Students join this kind of courses for many different reasons. It can be to gain personal knowledge, to progress in a professional career, to be prepared for a normal course. Therefore the number of registered users at the beginning of a course is significantly important.

4.1.2 Problems

As said previously, at the start of a course, many users are registered. However, only a few percentage of them will pass the course at the end of the period. This represents the biggest problem of MOOC, which concerns the retention rate of registered users [18, 24]. The reasons for dropout are many and various. The lack of time is part of the main reasons. The users can have time-consuming occupations such as a job, courses in a regular university. Another reason is the lack of interactions. For some users, a digital platform does not replace real communications with their peers. Moreover, with the large numbers of registered students, professors and tutors are not able to respond to all of the questions. One of the last reasons is the lack of background knowledge. As there is no selection to enter a course, some users may lack the basics to understand the content and do the exercises.

Another problem is the multiplication of platform used for communication outside the site even with the presence of a forum to ask questions and share information. Groups of students may create a Facebook group to share information [6]. Moreover, the lack of actual interactions with tutors or teachers is not possible due to a large number of registered students, and only a part of them interact on these platforms. The choice of tier platforms to improve the communications for special students (i.e., who paid tuition fees) may not be an ideal solution, bringing complication in the overall utilisation [29].

4.2 Mobile Applications

This part will show different examples of mobile applications which are designed for education purpose. They can be a stand-alone solution which is used for self-learning. Some of these applications are tools which can be integrated into a course to import new teaching methods.

4.2.1 Duolingo

The first example is one of the most popular mobile application developed to learn new languages, Duolingo. It is available in the app stores of the main platforms. This application was created to help individual learners who are motivated to learn a new language to start without previous knowledge. The business model of this free application is to provide translation for the web when users are more familiar with a new language.

The learning process is based on translation, which required the users to be fluent in one language (mostly for English speaker) to learn a second language. As shown in some studies [32], the effectiveness of such an application is mostly for beginners. For more advanced users, the progress is less visible in such a small period. No interaction is offered to users, or others type of content (e.g., video, newspaper), to learn with real case communications.

The paper [13] show the limitation of the level of users when they start with zero knowledge. The translations to be entirely correct need the input of more advanced speakers. This show the main limitation of this solution. Without teachers, and

based from the translation of a machine, it is possible to learn only basic grammatical structure which is not fundamentally wrong but different compared to a normal usage of the language.

The limitation that this application highlights as for MOOC the lack of possible interactions but mostly the quality of the content. On the contrary of MOOC, here there is no course inspire or created by well-known professors/universities. Therefore, it is not possible to expect to achieve the same things as with more traditional, human created courses.

4.2.2 Kahoot!

The second example is a web application which can be used either with a web browser or directly with mobile platforms. This application is Kahoot!. It allows teachers to create quizzes based on their courses [9].

This kind of application is used as a complementary digital tool to vary the teaching methods. Only the teacher needs to register on the application and create an account to add new quizzes. These can be shared with others on a public platform. To share a quiz with students, a teacher has to share a link with a code to be able to have access to this questionnaire.

To improve the interactivity in the question, different media can be added to the questions such as images, sounds, videos. After students completed all the questions, feedback is addressed to the teacher and also to the students. The names of the first five best students are shown, and the teacher can have access to the statistic of answers given in an excel file.

This kind of application increases the engagement of users during the learning process. The change of support make it more appealing, and the different sources of content allow them to be more critical of information found and then helps to find the correct answer, as a hint.

This kind of online quiz presents several limitations, concerning both of the content and its support. The first problem occurs when sharing the link to have access to a new quiz, which requires an external communication tool. Talking about communications, students are not able to interact, limiting the usage inside the classrooms. Finally, a limitation is in the quiz itself. Questions and answers have a length threshold, and teachers can create only multiple choice questionnaires with a maximum of four different answers. Therefore, its utilisation is limited to lower grades, to make the assignment part of the course more playful.

4.2.3 Padlet

The third example of applications is Padlet. Like the previous example, its usage is a complementary tool used during class. Padlet is a web application which can be used with mobile devices. This application consists of a digital wall where users can post different kinds of content.

This solution may be presented to enhance the collaboration between students and peer learning [12]. Instead of sharing direct vocal answers, every student in the

classroom can share content on a digital post-it in real time. The content can be plain text as well as digital content such as video, photos, hyperlinks.

This solution is based on the interaction for collaborative works, or to get a larger amount of answers, permitting shyer students also to give answers. It can be used only to provide a new digital tool to add interactivity during class; therefore it has to be integrated into a more global solution.

4.3 Virtual Learning Environment

Digital classrooms are developed to provide course content or tools used to replace or complete teaching methods.

4.3.1 Course Management Learning System

Course management systems are mostly used to share digital content [33]. These environments were underused, students taking the time to get more familiar with this new concepts. After some time, they feel more interest in what they are learning by using this method and also start to provide feedback to improve the content provided [21].

A famous example is the open source solution Moodle. It is used by many schools and universities to provide extra content to students and as a communication tool. Content can be either lecture notes, or home assignments (such as quizzes) [21]. Information of a course is divided into different blocks; generally, the first ones correspond to general information on the course, and then it is split by one block per chapter. It is integrated into courses to provide additional content, or for independent work for students.

4.3.2 Google Classroom

Google classroom[4] is developed to integrate the different Google services inside the classrooms. Thus, teachers can share all kind of content by using Google drive. Its integration with the drive of students automatically creates separate documents for personal assignments. Moreover, with docs, it is easier for students to study by groups with collaborative work tools.

As it used Google applications, these services are SaaS (Software as a Service), that means it corresponds to a cloud-based solution, available everywhere. Also, these applications are well integrated with main mobile platforms (especially for Android).

The main drawback is the lack of courses. There is no additional service which was added to create new courses. This solution can only be used to deliver extra content for students.

[4] https://support.google.com/edu/classroom/answer/6020279?hl=en

4.3.3 Apple Classroom

Apple classroom[5] is an application to provide courses. The teacher can share their different courses with their classes, and divide these classes by group of students. It is well integrated with other Apple features, for example with airplay, to share the screen of the iPad of the teacher, monitor the different devices or display one student screen on an Apple TV.

This application is only available for Apple devices which have the latest version of the operating system. Therefore, this solution is not cross-platform, and lack of compatibility with older devices. That is to say, that for intensive use in the classroom, schools have to provide the devices to the students.

This solution has some limitations concerning the technology used and the content shared. A Bluetooth connection is required to connect devices together and make the link between teacher and students. However, if a student disables it, a teacher will lose access to screen monitoring or the ability to lock the device. Another issue is when a teacher wants to use an application or a book as content for their course, they can trigger the same action for the devices of their students. However, if the app or the book is not present on the device, there is no way to automatically provide the content to the student which will be, for the best case, redirected to the store.

As this solution is presented, it becomes beneficial as a complementary solution with mobile device management. As shown in the Section 7.5, it is the solution chosen by some MDM providers.

4.4 Drawbacks

These current solutions used for mobile learning do not cover the different challenges listed in the previous section.

4.4.1 Problems not Solved

The first issue of all these possible solutions is the lack of complete products. To obtain a fully functional system, it is needed to combine several products together. It is, for instance, the case with Apple classroom which only takes care of the course management. Therefore, the users need to manage a user base simultaneously on, at least, two different product. It means to double the end-users management and also mixes the user experiences, making harder to learn and adapt the ability of end-users. Therefore, it increases the complexity of the global system, reducing the interest of such solutions.

The second issue is the few interactions between end-users which are provided in these solutions. Solutions are not using the full potential of mobile devices, only providing media content to students. Mobile devices should be used to permit better group work organisation, direct communication with the teachers.

Some of the solutions described above are developed only for one platform. Thus, it forces schools to ensure that students have the right kind of devices to have access

[5] https://help.apple.com/classroom/ipad/1.1.1/

to the content, preventing them from having a BYOD policy.

4.4.2 Implementation Criteria

From these problems, it is possible to draw a list of required criteria which are solving the issues previously mentioned. They will define the needed implementation, to propose a complete solution. These criteria are divided as follow: simplicity, compatibility, automaticity, controllability.

The main requirement for the implementation is to have a global solution so that the system is managed at a central point. It means it has to include users and devices management as well as integrate courses sharing tool. It should contain all the needed part of a global system, but not necessarily all the possible services. Thus, it should be compatible with others services and the different media content, for instance, content sharing services. That means, there is no need to recreate multiple platforms to recreate tools if it is possible to integrate them (for example, like cloud storage, with the integration of Dropbox, or Google Drive).

This compatibility is also necessary for the interoperability of the system on different platforms. The system should be available on different OS. It will have to permit a maximum kind of users to have access to the system with their device and enjoy its capabilities. This compatibility concerns the system management and the course content. Meaning the system have to provide similar functionality independently on the Operating System. Moreover, it should avoid using a proprietary format as a key feature for sharing content for a course.

The main criterion when using mobile devices concerns the applications access. Thus, the application manager is necessary to ensure the required apps are always installed on every device. It can also be used to share a custom application of the schools to end-users (i.e., teachers and students). This application management needs to be automated, to facilitate its utilisation. This automation can come from the creation of tasks directly on a server, or with the range of actions to multiple devices at the same time.

Finally, the solution needs to solve the question of mobile devices source of distraction, which represents one of the concern when talking about mobile learning. The solution should provide tools to ensure a better productivity when learning by using a device. It can come, for instance, from restricting the utilisation of the device, or blocking its access.

5 Mobile Device Management

To address the problems of mobile learning, Mobile Device Management is chosen as a basis to develop the system. The technical implementation of the solution will be detailed in the Section 6.

This section will define the main features of an MDM solution. Firstly, MDM is presented as well as current related work on this topic. Then, specifications of the main features are shown as well as the developed solutions of the two most important Operating Systems for mobile devices.

5.1 Definition

Mobile Device Management is used to administrate devices, such as smartphones, tablets, or other computers. This solution is used for organisations (education or business) with plenty of devices needed to be managed. MDM allows managing policies, installed applications, stored data and services [26].

Policy

By changing the policy on the device, it is possible to enable or disable different features. That means policies set up rules for the device. For example, with an MDM solution, the applications store of the platform can be disabled. Thus, a user is limited to a predefined usage by the company. For privacy issue, sensors like the camera can be removed from the device.

The interest with an MDM solution is the possibility to apply different policies for different groups of users. A policy can be applied depending on either characteristic of the device, such as, its platform or if it a smartphone or a tablet, or the role of the user assigned to this device, who is registered in the system.

Application

With an MDM solution, the server can decide to install or uninstall applications. This feature is possible only for managed applications, meaning install by the MDM process. However, it is possible to change an application to a managed state.

The AirWatch application management specifies three different areas. This MDM solution offers their clients to distribute, secure and track applications. The distribute area corresponds to the installation process. Like others MDM product, it is possible to install a specific application, automatically perform installations during the enrollment phase, and also give access to a specific enterprise catalogue. The secure domain concerns restrictions on applications. It can filter application installation from the store, or even disable official stores. Moreover, it is possible to restrict the utilisation of pre-installed application (meaning, installed before the enrollment). The tracking of applications allows the client to get a list of installed applications on the different devices, as well as log when a new application is added to one device.

Data
One of the major concern when using an MDM solution is to be able to protect the data from the company. First, the MDM can be used to provide access control with trusted devices when accessing to internal servers. In that case, the MDM server will add the required authentication information during the registration phase. If the device is lost or stolen, the MDM server might send a wipe control to reset the data stored on the device.

Services
The services regroup several features an MDM can interact with mobile devices. One of the main functions used is the localisation of the devices. It can be used to find a lost device, and also to restrict the utilisation of the device depending on trusted places. If we take the example of Apple devices, the MDM solution can be used to control the AirPlay system, which can display the screen of a mobile device to another screen, such as, an Apple TV.

MDM was first developed for business enterprises. It is now, adapted to another kind of infrastructure, like for example machine to machine communication system [8]. Currently, solutions of this sort are also used for education. Indeed, with the democratisation of mobile devices, schools and universities have started to get similar projects [27]. Part of the services offered on campus is based on the digital solution. Indeed, with the different kind of actors, MDM can alter the view depending on the role of users.

5.2 Related Work

This part will mention the research work related to MDM. Most of these works are focused on security and privacy issues. They can be related to network security but also from users perspective with, for example, the BYOD policy.

These topics will not be discussed in the implementation which will focus on features for mobile learning. However, these concerns have to be integrated into the development of any MDM solution.

5.2.1 Bring Your Own Device

Mobile Device Management is often tight to Bring Your Own Device paradigm. Indeed, with a BYOD program, users can possess a large variety of devices, different from each other.

The main difference is the platform used by these devices. Two different Operating System for mobile devices do not work the same way. They have two different set of applications, link to their applications store. Capabilities of devices may also differ, depending on the supported version of the Operating System.

The difference can also come from the hardware, smartphones and tablets may be used for different purposes, and some inbuilt features may not be present for every of

them. Sensors like camera, GPS can alter the utilisation of some services. Therefore, MDM can apply restrictions to adapt the services to the capacities of the device.

Moreover, MDM can be used for authentication to access secured services, using trusted devices as identification requirement [14]. Enrolled devices can then have, for example, authentication token, to identify its user. By managing the data locally stored, an MDM can ensure, and privacy and the integrity and content shared and stored. Based on the device policy, MDM solution can increase the security level of devices [14]. Security features like auto-lock or password length can be automatically changed.

Studies are agreed to say that BYOD programs have a positive impact on both employees and employers. Moreover, it requires adding a set of security and privacy measurements to protect the system of the company [23].

5.2.2 Security

Security is one of the most important parts of research concerning MDM. Indeed, it is important for such crucial system to know where may be the weaknesses. For this, researchers used threat modelling to highlight the assets of these systems and to discover the nature of these threats.

The assets consist of the control of devices and the access to confidential data of enterprises. The attacks against the data may come from either the users or their devices (i.e., independently of the will of the user) [26]. This is why groups of users are created to give more or fewer rights depending on their role. They are divided into, at least, two categories which correspond to end-users and administrators.

These authors of [25] propose a list of threats to MDM system. They added possible solutions and requirements to ensure the security of the system. Devices can perform attacks made by the user itself or by a tier person who had taken control of this device. The first threat is someone, with potential malicious intentions, which may disclose sensitive information. Most of the time, the device can be used for authentication to access data servers. For better security results, it is advised to have segmented users as well as two-factor-authentication.

A secured connection is recommended to ensure the privacy of communications. Moreover, encryption for local storage might be required to protect the data when the device is lost and offline, i.e., when it is not possible to send a wipe command. The encryption of connections, while the MDM server sends commands, is essential for the integrity of devices. Thus, it is important to protect messages to avoid their alteration by an external actor (such as, with man-in-the-middle attacks). Depending on the policy, MDM system may have much control over a device while being managed. Mutual trust is needed during connection establishment. Threats can compromise either the devices or the infrastructure (like internal servers) of the company. Malware can be installed for data gathering or to create a botnet by controlling devices.

The paper [14] explains the different kinds of policies concerning device ownership in the company. They highlight four different strategies, which consist of: Here is Your Own Device, Choose Your Own Device, Bring Your Own Device and finally On Your Own Device. As explained by the authors, except the last policy, the company

can manage and control devices. That is to say, thanks to an MDM solution, the company can ensure the security and the privacy of its data and the integrity of its system. This control over the device is defined by rules that the users have to follow to meet the enrollment requirements [14]. As defined in their paper, these rules can correspond to force users to choose a strong password by setting requirements, like certain kinds of characters, minimum length and so on. It can also manage data by doing backups or wiping data from compromised devices.

The policy rules that are used by an MDM solution correspond to the minimum security rules needed to communicate with the server. Therefore, in the case where several policies are installed on the same device, each rule will correspond to the stricter one.

To be able to perform these different actions, the MDM server uses a set of API. Depending on the platform, the way of managing commands will be different.

5.3 Technical Specifications

This part presents the specifications described by the Open Mobile Alliance (OMA), and the ones of Apple and Google platforms. These technical specifications will be used to implement the MDM part of the solution presented in the next section. Only the Apple and Google platforms are taking into account because they represent more than 95% of unit shipments[6].

5.3.1 Open Mobile Alliance

The OMA is a standards development organisation which develops an open standard for the mobile[7]. In their standard, they worked on Device Management protocol [3]. Before able to start a management session, the client needs first to be provisioned with a bootstrap process [2].

Bootstrap

The first step to integrating a new device to an MDM system is the enrollment phase. The device and the MDM server have to communicate to link the device of the user to a logical user/entry in the system. The OMA specification identifies four different methods to initiate the creation of the device in the MDM system.

The two first possibilities correspond to pre-registered methods. Pre-registered means an entry is created before the activation of the account into the system. They consist of a factory bootstrap and the usage of a smart-card. For the first one, the client is provisioned with factory setting, that is to say, the client is built-in at the source of the device, as a by default service. With this solution, the initial configuration is kept even after a factory reset. The second solution is by using a smart-card which contains all the information needed for the provision of this device. Each smart-card is pre-integrated in the system before its utilisation.

[6] http://www.idc.com/promo/smartphone-market-share/os

[7] https://en.wikipedia.org/wiki/Open_Mobile_Alliance

The second type of method is network based. Either of the device or the server initiates the connection for the bootstrap process. In the case where the server initiates the dialogue, the user has to give an identification, such as a phone number, to the server to be identified. If the communication starts with the device, it has to have the address of the server, for instance, received by email.

The bootstrap process will provision the device with an initial profile to be able to start managed session with the server later on. On the contrary of future messages, during the bootstrap process, the server send a one-time message without excepting any answer.

Management Protocol

After a device has been registered to an MDM server, it can start to be managed during management sessions. The necessary commands used for this management are sent by messages through the network. As for the provision part, these messages follow an XML-based format for full compatibility.

The most important rule concerning the messages sent is the limitation of messages that can be sent at the same time. As it is not possible for the server to know the quality of the network or the availability of the device, it can send only one command at the same time. The server needs to receive an answer before sending the next message, that is to say, the next command. The only exception is when the server is sending large objects which required several messages to be transmitted.

To initiate a new managed session, the server will send an alert message to notify the device a new command needs to be performed. Then the client will respond to the server by sending information to identify itself. Therefore, the server will be in measure to send the corresponding commands. Finally, the client will send the status of the execution of the command to the server which will terminate the session or send the next command to the device.

This scenario happens when the device is online and available to perform the action. There is a possibility for the device to receive a command that it is not ready to execute. In that case, it will send a 'Not Now' answer to the server, to indicate to this one, it will need to send it back later.

It is also possible for the user to respond to commands, to interact with the server. A message that contains a command consist of the id of the command, data depending on the kind of command, as well as optional additional parameters. These additional parameters are used to adjust the time the message will be displayed, and to control the response of the user. The kind of answer can correspond to yes or no questions, accept a pop-up window, respond to a multiple choices form, or this user can also enter different kinds of inputs. The input is limited by the additional parameters sent by the server, which specify the length and the kind of expected data (e.g., characters, digit, date, base-64 coded file).

Security

Security key points are also defined by OMA [4]. In their standard, they start to show the different types of credentials that can be used between an MDM server and a client. This credential can come from the server or the client with a couple

username and password. They may utilise as well, a certificate, or a communication mechanism (like a particular type of network, or layer).

To start a new session, either the server or the client sends the credentials (most of the time, username and password) or ask the other one to send them, in response to the initial message. To ensure the confidentiality of these sensitive data, it is advised to use a secured connection. In the case where the server sends an alert message to initiate a new session, it may not have the credential, as it sends a simple notification message. Notification messages are used because the device may not listen to others channel for more protection. For this scenario, it is important to ensure the integrity of the message, by adding a digest string to the message (i.e., by adding a checksum like with MD5).

In addition to secure management sessions, it is crucial to ensure the security during the bootstrap phase. Indeed, at this moment none of the two actors had already start communication. Thus, the client and the server have to be able to trust the other one during the first exchanges. When the bootstrap process is initiated with a smart-card, then it is possible to use pre-defined encryption. However, for network initiation, an important step is to confirm the integrity and the authenticity of the first bootstrap message. Therefore, it is possible to use an id from the network and/or a user pin. The choice of a user pin has to be made between the user and the server before the bootstrap process, by using another channel (i.e., using e-mail).

5.3.2 Apple MDM Specifications

Apple devices are the main focus for MDM provider specialised in education.

Check-in and Enrolment

To enrol a new device, the server first creates a new entry and push to the device its authentication payload with the corresponding token. The payload message consists of an id for the payload, the URL of the server, and the check-in URL.

When a device received a new MDM payload, it will check-in to the server to identify itself and send initial detail information. Among this information, the server will receive the operating system version, the build version, the serial number and others code of identification.

The device also sends a message when it needs to change its authentication token so that the server can continue to send commands and messages to this device. For both check-in and token update, the server has to send back an acknowledgement message with a status. This status can be either 200 if everything is OK or a 401 error, meaning this device is not authorised to connect to this server. In the case, of the error message, the device will automatically remove the MDM payload, from its DM management.

The last case of messages sent by the device is the checkout messages. After a device is unenrolled, it will try to send only one message to the server to indicate it will stop to be managed.

MDM Protocol

The server also uses payloads, to send commands to a client. The command payloads have two more fields: the id of the command and the dictionary of this command. In the dictionary, the information contains the request type, to identify which command needs to be performed.

If the device is connected, it will send a response to a command payload message. Its answer contains the id of the command, the id of the device and a status resulting from the command (an extra error field can be added). The device can send a 'Not Now' answer if it can not perform the action immediately. In that case, the server has two possibilities: send another command and retry after, or it can stop the session.

Apple allows using only pre-defined commands with an MDM solution. They are used to manage the profile installed on the device, to install or remove applications, change security settings (like resetting the password), lock or reboot the device, manage the data (e.g., wipe and do a factory reset), or use some services like mirroring control with AirPlay.

In the case of managing the AirPlay service, the server will send a command with the request type RequestMirroring. In the dictionary of this command, it is also possible to add some optional parameters which can be required to start a new mirroring. These parameters will define which device will be used on the local network, as well as a password if the access is protected. It is also possible to specify the duration during which the device will scan the local network to locate the desired device (such as an AppleTV).

Supervised Device

It is possible with Apple devices to supervise them. Supervision improves the control over devices and permits to set up more settings for the policy of devices. For instance, it is possible to add more restrictions compare to normal enrolled devices, like fixed lock screen and background. It is also possible to add a network configuration, like adding VPN information.

It is possible to supervise any device by using Apple Configurator. It is also feasible to use the Device Enrollment Program (DEP) for new devices. With DEP, Apple directly bootstraps the device with the company configuration. The DEP method is more efficient to configure a complete fleet of new devices.

With DEP, an institution can directly integrate the devices into its he MDM solution. The profile is pre-installed, limiting the action required to be done by the user during the first utilisation of his new device.

5.3.3 Android Device Administration

This part will present how Google is handling Device Management for their products.

Android Device

Android 2.2 introduces support for enterprise applications by offering the Android Device Administration API. To start using a management solution for Android, an admin application requires running as device administrator in the device. This

application can modify the policy installed on the device.

To distribute this application to the different users, several strategies can be used. The easiest one may be to provide the app on the Play Store (official store for Android application). For this scenario, the end-user just has to install the application like any classic app. Another possibility is to send the application on the device, by downloading it, from a server, a local machine, or sending it by e-mail. Then, the user will have to enable the installation of tier local apps before installing the app on the device. Finally, the user has to activate the app as administrator to be able to use its Device Management functionalities, by accepting the prompt window at the end of the installation phase.

The application will install the policy needed to have access to the Device Management server. As for other MDM server, these policies can control, for example, the requirement of passwords, parameters for the device lock.

On the contrary of Apple API to manage devices, there are no pre-defined commands, for instance, to manage the installed applications. In the case of Android devices, it is required to develop custom commands by using the admin application, which is already installed for the enrollment. System commands can be used from the application to perform the different operations. Thus, a custom API has to be implemented. This allows more possibilities, like for example, for the installation of applications. It is, in the Android case, possible to install the application not available on the public market, such as special enterprise apps. To do so, it is needed to push the APK file in the local storage of the device, and then install this local file.

Chrome Book

In a future version of Chrome OS, it will be possible to have access to the Play Store, that mean using Android applications. Chrome Books represent 60% of the sales of computers in the US for k12 students[8]. This information makes Chrome OS a good candidate for educational, with similar usage as Android devices.

The enrollment phase for these laptops is different from mobile devices. It is required to register the device to an educational Google account using the Google services. Then, the manager account gets the list of associated devices to an account through an API. Moreover, others command also require to use the Google API.

Enrolling Chrome Books is useful to authorised these laptops to connect to a private network, or also install required applications. Even if some possibilities exist, there are not as developed as for mobile devices, meaning less configuration compare to a custom API for Android, or fewer services compared to Apple devices.

[8]Google I/O 2017

6 Technical development

The following section will demonstrate how Mobile Device Management can be integrated into an educational application. The focus will also be made on the light clients designed to improve the ease of use of a mobile solution. As previously shown, MDM is only needed for practical features to enhance a broader system.

The application is developed mainly for elementary-school, mid-school and high-school. Thus, the design of the implementation takes care of the importance to simplify the utilisation. This simplification is adapted for kids as well as teachers so that no technical people are needed for daily maintenance.

The presentation of the implementation includes the following parts: user management, Device Management, application management, and finally, course management.

6.1 User Management

The user management focuses on the development of a single user system. The infrastructure is implemented to regroup all users in the same instance. With one system to manage the users, there is no need to register users for each of the services used, based on systems with different mechanics.

The system defines specific roles for each kind of users, to avoid to mix different action, to reduce the complexity of the system.

6.1.1 Unique System Infrastructure

The solution is hosted in the cloud. All the services provided are web-based. The first impact is no local storage on the mobile device is used. It allows more variety of devices as it is not platform dependent. Moreover, shared devices can be used, as the content of a user is not based on the device used.

The infrastructure is divided between the back-end and two clients: one mobile with the application and one on the web with the web-console. The backend is mainly used for the Device Management features presented in the previous section. The user interfaces will be described later on. They are used for the course management and to configure different part of the solution.

The solution is divided into three layers. The first one is the system layer. Then, the two others consist of the instance and the school layers, which are controlled by the customer.

System

This solution is host in cloud servers. The system level corresponds to the global state of the solution. It contains the different instances.

Several different systems may exist. All of them will have a separated database, thus are completely independent. As it is a cloud solution, one system may be hosted on different servers, to balance the load from the different customers.

As this level is more general, it is not used for the end-users, nor the customers. One instance is created per customer; hence, all instances are independent of each

other.

Instance
An instance is the first level for a customer. It allows creating all the structure of objects needed to start using the product. It is used to have access to the web console.

This layer is designed to manage the different schools owned by the instance. Therefore, it is possible to set default parameters for policies and application catalogues. For these options, it is even possible to prevent modification of some values at the school level.

With instances, administrators can manage applications, users, devices, schools. However, they have no control over the learning contents. They can only interact with the Device Management system.

Specific values, such as the update time specified to update the device information, or the list of installed application can be set per instance. This time is calculated in function of a base time, generally one hour, and a coefficient. This coefficient can be changed per instance; moreover, if no value is specified, then a default system value is selected instead.

In one instance, it is possible to have multiple schools. There is a relation of inheritance between an instance and the different schools that are maintained by this instance.

School
The school is the last layer in the system infrastructure. It focuses on the learning management even if some Device Management features subsist.

If the parameters are not protected by the instance, it is possible to adjust them at the school level. The parameters are set for device policies and applications catalogues. Therefore, a school can get its own VPP applications (explained in Section 6.3.3), using a token different to the one that may be provided by the instance.

The school level also adds the creation of classes which bring the notion of mobile learning to the whole solution. The learning management mostly consists of the course management which will be explained in another part. As this layer is used for mobile learning, this explains that it is also utilised for the mobile application. Indeed, each session use to connect to the application corresponds to an end-user (e.g., student or teacher); thus based on the school layer.

Each of the layer described above also gave access to some services. Credentials used for authentication can be stored either at the instance level or the school level. Therefore, it will authorise the access to the services registered to the users belonging to these groups.

6.1.2 User Roles

This part shows the different roles of users which exist for the system. They are divided into two groups: administrators, and normal users.

Admins

It exists one type of administrator for each layer. The admins are composed of sysadmin for the system layer, instance admins for the second level, and finally, admin teachers for the school layer.

The system admin manages the whole system. Thus, this user is used to create new instances by setting up the instance and the first instance admin. He participates on the initialization for new customers.

From this point, all the other parts are done by the instance admin. This user will create the other users of the instance, such as other instance admins or teachers. He will also create the different schools. An instance admin can not use his session to connect to the mobile client, as he has no role in the course management.

Admin teachers have the same level of control as instance admin, but they do not have access to instance settings. They can manage details of teachers who are registered in the same school. Moreover, they have the same possible actions as normal teachers, who are defined in the next part.

End-users

End-users are the ones who are using the mobile applications. They are not involved in the settings of the system and mainly use the course management services. Their global interactions with the system are limited.

The first users are the teachers. Unlike the admin version, they can only modify information related to them, such as editing their information or managing their devices. They can create courses, and also manage students.

Students are the users with the fewer possibilities. They are not able to edit or manage their own devices. They can only use the mobile application and have access only to courses where they are enrolled.

End-users belong to one instance and more specifically to a school. That gives them access right to features included in their school or their instance. It means they may have access to others services managed by their group. Therefore, with this end-user management, it is possible to associate several services to a group of users, to have a complete solution.

6.1.3 User Interfaces

This system is controlled by users through two different interfaces: a web site and a mobile application. Their development and design were based on facilitating their utilisation for a majority of users. The mobile application is used by kids, so it requires a light and efficient interface. It is also valid for the web-console which needs to be used by any teacher even with few digital skills.

Moreover, the main aim of the global solution is to enhance the ease of use of everyone on their own task. It is developed following the cloud philosophy: hiding the complexity from the users. That means only the needed information for configurations are asked to the users, the MDM server taking care of the rest.

Web-console

The web-console represents the main interface with the system. It is from there that users can manage their instance.

To log in, a user has to provide the instance name as well as his credential. Then each different section is divided into several tabs one per theme. Each view displays the set of possible actions and clear information. It is so possible to have access, depending on the authorization of a user to the devices list, classes, students, teachers, schools and instances.

This web-console also provides some statistics about the system. Statistics is also an important information required to monitor an instance easily and to provide a clear overview. They can be at different levels, such as instance, school, device, course. For the instance, it gives the number of the user registered. For the school level, it indicates the number of active devices as well as users who may use the mobile client.

To keep a trace of the activities, the web-console also give access to the logs of some operations. As it was explained for devices, to have a visibility of the actions done by the server, it is important to be able to follow the main operations. In the case of devices, it can be, for instance, to control the installation process. If an installation failed, it could indicate if it is a communication problem, with a pending installation status, or if it completely failed. Moreover, installing many education applications takes a huge amount space; thus indicating the available storage information helps for an easy diagnosis.

Mobile Application

The mobile application is based on web-view to display content to end-users. It permits better compatibility of the code between the web-console and the application. Moreover, the website system allows getting the same features for all the platforms with similar code, increasing the interoperability, by offering the same experience for all the platform.

This mobile application has two different views. The first one appears in the case the device is not yet enrolled in the system. This first view will guide the user through the different step required to be done for the bootstrap process. It, for example, provide the Qr code reader to read and decrypt the token generated by the web-console during the enrollment process (explained Section 6.2.1).

The second view corresponds to the mobile learning part and gives access to the different courses. This part replaces the enrollment view automatically, as soon as the MDM profile is installed and the device check-in to the server.

After that, the client is directly linked with Device Management features. The first performed action is the activation of the localisation (if enabled with the policy) when someone run the application. Moreover, the Device Management simplifies the login process when using a device assign to an end-user. However, it is still possible for a user to share his device by login out. Then, the other end-user has to use his credential. As the solution is cloud-based, he will find his session as if it was his own device.

From this client, it is possible to get the same application list as for the device

detail in the web-console. The end-user can know which applications are installed and managed by the MDM. Moreover, he can also request the installation of an application directly from this list.

Teachers can also manage the devices of students that are currently following the course. This management includes the list of active students, and from this list, there is the possibility to send a lock command simultaneously to all their devices. This feature is important in the case of mobile learning to decrease the possible distraction created by mobile devices and keep a certain control inside the classroom.

The distraction is one of the main concern before starting using mobile learning. This is why, Device Management bring value, especially concerning supervised devices. Depending, on the location or the time, if a student uses the device during a course; the policies can evolve with the environment. Restriction of content for the internet, disabling the store can be pushed over the air.

In other words, when using the applications during classes, a temporary policy can be pushed to the device. Then, the required restrictions are applied, to limit the utilisation of the device only for the course content and applications included for this course.

6.2 Device Management

One of the most important point concerning devices is the support of different kinds of model. Indeed, concerning schools, it is important to support the difference in projects, based on different platforms. For example, two different schools, may not provide the same tablets. Moreover, this is emphasised with the BYOD paradigm, in this case, it is possible for students to use their personal devices. Thus, different policies and flows have to be developed to take into account the majority of possibilities.

6.2.1 Enrolling

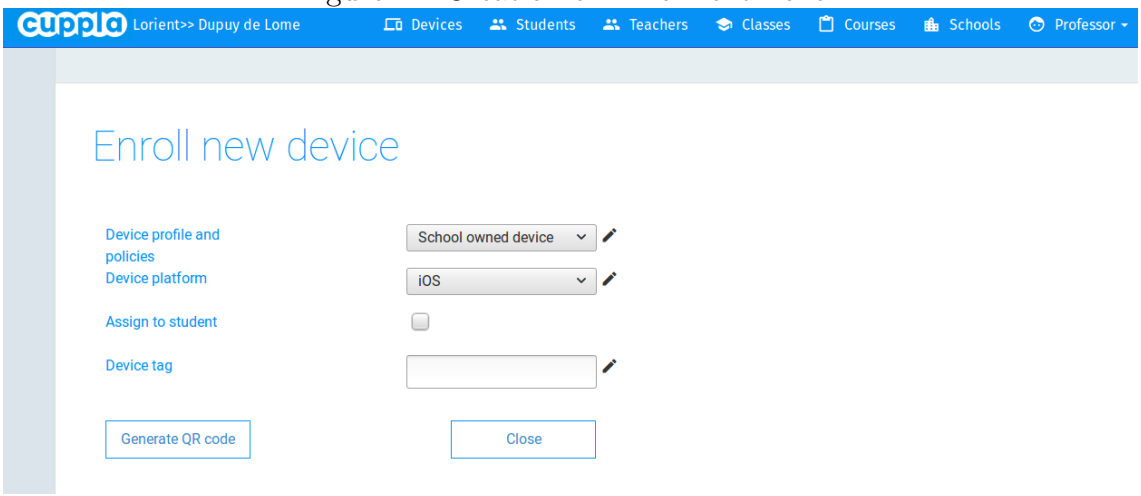
The enrollment process corresponds to a network bootstrap process. The enrollment is the crucial moment for an MDM and sets up the trust between the device and the server. The Figure 2 shows the enrollment steps explained below.

Token Creation

During the enrollment phase, the type of the device needs to be selected. Three different kinds can be used to characterise a new device. These types consist of School Own Device, Student Personal Device, and Teacher Personal Device. School Own Devices are the ones which are belonging to the school and shared with either students or teachers. The Student and Teacher Personal Devices concern the BYOD policies.

The enrollment flow starts by generating an identification token. It is produced with a randomly generated number and options selected by the user. Among these options, the user has to determine the kind of device, choose who will use this device. As shown in Figue 1, it is also possible to add a tag, meaning a personal identification for this device. This information will be associated with a temporary token.

Figure 1: Creation of Enrolment Token



The token is presented as a QR-code. It simplifies the connection process, reducing the steps the end-user needs to do. Part of the information required (like the server address) is hidden from the user and directly managed by the server during the creation of the token. By scanning the code with the client application, the user does not have to enter the device type or the server address.

Bootstrap Phase

After downloading the client, the end-user will be able to start the enrollment by using this token. With the token, the device will be able to authenticate itself during the first communication with the MDM server. Then, this server will register this new device using the first entered information during the token creation. When a new token is used, it will raise a flag to avoid two devices to register, using the same token, and so, ensure the control on the server access. A token has also an expiration date to prevent token to be used if they are disclosed (for instance, end-users forget to disconnect from their session).

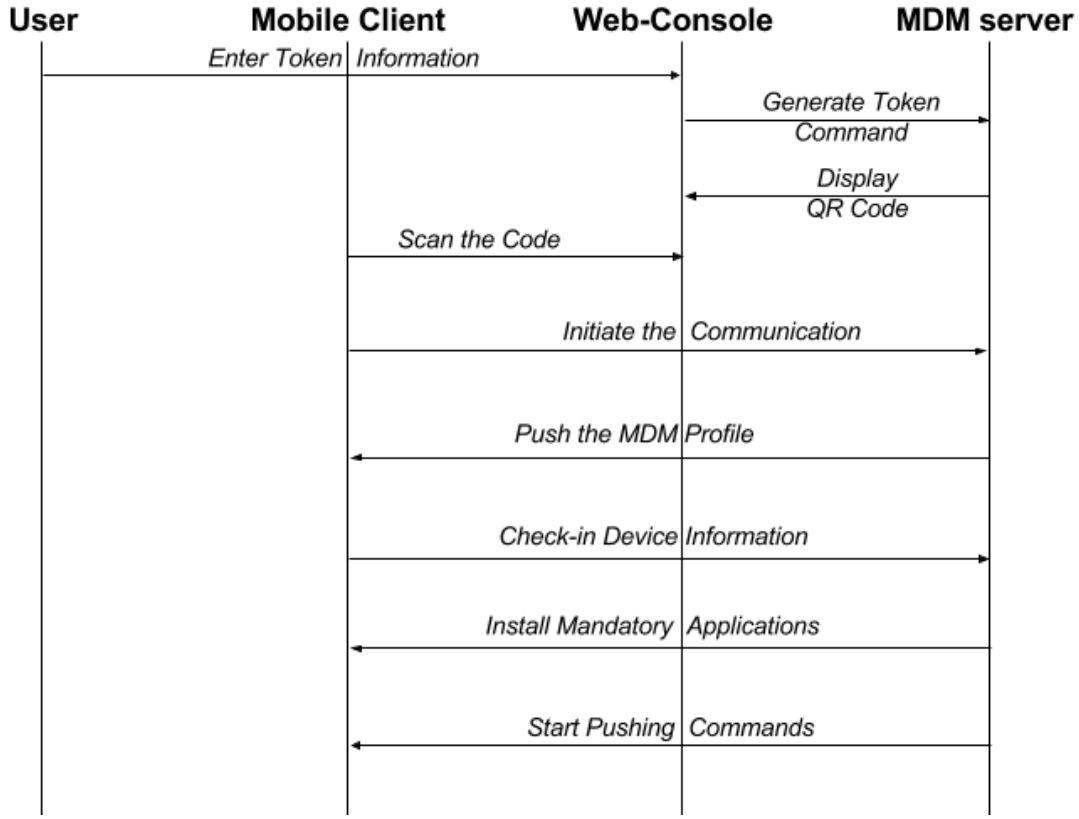
Each new device has three different ways to be identified using id. From the database point of view, with its public key. For the user interface part, it is possible to identify it by a unique ID for each school, automatically generated by the server. For the user point of view, by the tag (editable) previously chosen, which can be associated to a sticker on the device.

After the enrollment, depending on the policy, the users may have to do some modifications. The new policy may require enforcing his password if his current one does not fit the new security rules. Settings about the time before locking the device or the accessibility to the store may also change during the enrollment process.

The enrollment can fail if the device is already managed by another system. Indeed, only one MDM solution can control the same device. In that case, the user has to remove the previous profile and only need to redo the previous step, i.e., installing the new MDM profile.

Check-in
After a new device is enrolled, the user has to go back to the client app to finish the registration phase of the device. At this moment, when the user launches the app, the device check-in to the server by sending some information. This information corresponds to the serial number, the model, the operating system version, the amount of memory.

Figure 2: Initial Communications



At the same moment, the server pushes the installation command for mandatory applications (i.e., which are required to be installed on every device). It ensures to get the device in an initial state that fit all the needed requirements.

After the enrollment, if a user is assigned to a device, this user does not need to authenticate himself to log in to the client. If the token of the device is still valid, it is used for the automatic connection into the application.

Un-enrolling
The device is enrolled in the system until it is unenrolled. There are several scenarios when removing the MDM profile is needed. The first case is if the device will not be used inside the system anymore. The second scenario is for devices that have not connected for a while. For both cases, it is important to manage only active devices, to keep a relation of trust with all the devices.

It exists three way to unenroll a device. The first possibility comes from the end-user because it is possible to remove the Device Management profile from the settings. The two others possibilities come from the MDM server. The first case is if someone performs the un-enrolment action from the web-console. The second case is an automatic unenrollment after the device has not communicated with the server, after a defined period. This last case can be used to ensure the security when a device is lost or forgotten.

To auto un-enrol devices, a task is running in the background of the server, and regularly check if the devices that haven't tried to connect to the server during a delta time. If so, the server will call the same command has the web-console action.

After a device is un-enrolled, the next time it will try to start a new connection with the MDM server, it will receive a 401 status code, as an answer, instead of a 200. That means it is not authorised to access the server. Therefore, the device will automatically remove the installed management profile.

When a device is removed to the system, the policy is removed, and restrictions are deleted. Moreover, the managed applications by the MDM are automatically removed. Thus, the device will go back to its previous state. In the case of Android devices, the device administrator service has to be disabled first, before uninstalling the mobile client.

When un-enrolling a device, the information related to its user is deleted, for privacy concerns. However, it is possible to reactivate a device by doing the enrolling process again. Therefore, it will keep part of its information that was previously stored.

6.2.2 Command

To interact with devices, the MDM server is sending commands. These commands can be used to gather information or to perform actions on the devices. Examples of actions are presented in the following part. They can be created by the server with a routine, or initiated by the end-users. Routine consists of a process which will execute a list of tasks at a fixed time or after a certain amount of time.

The process to send a command to a device start by creating a new command object. It is created for a particular device and is initialised with the needed parameters. These parameters can be, for example, the ID of an application. Several commands can be created at the same time and queued in the system waiting for the device to connect to the server. Then a push notification is sent to the device. If this one is not idle or busy, it starts a new connection with the server. During this connection, the device asks the server what it needs to do, meaning if there are commands queued for it. After that, it executes the different commands. For each one, it sends back an acknowledgement to the server to indicate if it succeeded or failed. In the case the server received a failed acknowledgement, it will attempt later on two other times.

To keep a trace of the activity of the different devices, each time a new command is created, a new entry in the log of the device is added. A second entry is added when the device sends the acknowledgement. These logged operations can be used

after to diagnose issues.

The commands that are created by the server are used to update the information about the devices, to keep them up to date. It concerns, for example, the battery level or the remaining memory. If a device has not been updated in a certain delta time, then a new command is created, and a push notification is sent.

However, to avoid to overload the server only a bunch of devices is checked at the same time. To compute the number of devices that can be updated at the same time the delta time is dived by the time of the routine to know how many time the action can be done (equation 1).

$$number\,routine = \frac{delta\,time}{time\,between\,routine} \tag{1}$$

Then, the total number of devices is divided by this number to get the number of devices per bunch. It is important to add plus one device per group, to ensure to update at least one device each time if the total number of devices is too small (equation 2). Indeed, if an instance has few number of devices and the number of time is executed during a delta time is too important the previous formula may give a zero. Thus the importance to ensure at least one device is selected every time.

$$device\,per\,bunch = \frac{total\,number\,of\,devices}{number\,routine} + 1 \tag{2}$$

To avoid to take always the same devices, as soon as a command is created for a device, its timestamp for this command is updated. Thus, even if this device is offline, it will not be chosen again in the next batch. Moreover, devices are sorted by their timestamp always to take the oldest first.

6.2.3 Services

The services supported by the system correspond to an action that a user can initiate from the web-console to interact with a device. When a user wants to activate a service, it will create a new command and thus follow the same procedure as described as in the command part. Therefore, services correspond to the possible actions offer to end-users to interact with the Device Management system.

Here are described the features provided by the system. These actions are related to general MDM functionality. The application management is also part of the available services; however, it will be described in more detail in the next subsection.

Lock

The lock feature represents the most interesting interaction for education. One of the

main concern when talking about digitalisation of schools is the distraction generated by the devices. It is so important to allow the access of the devices only when needed.

Therefore, it is possible for the teachers to lock devices. From the web console, it is only possible to interact with one device at a time. After the user clicks on the lock button from the web-console, the page is refreshing waiting for an answer. If no acknowledgement is sent from the device to indicate the status of the command, a timeout is done for the page refreshing. On the contrary, from the mobile application, this action is sent to all the devices of students who have entered the course. The course management will be explained in the next section.

Wipe

The wipe action, unlike the lock, is only available from the web-console. Moreover, no answer message is expected for this command. When the device received it, it immediately does a factory reset.

All data locally stored on the device is removed. It is so important to protect this functionality from miss-click. To do so, two confirmation message is shown to the users. The first message asks the user if he desires to continue or cancel the action. In the case of a positive answer, the confirm button trigger the next message. For this second confirmation, the user is prompt to enter a randomly generated number. To, create the command and send the push notification to the device, the number enter has to be similar to the one shown in the alert box.

Assign

School Own Device can not have user affected. When the token is created, it is possible to assign or not a user already registered in the system. If no user is assigned to a device, when the client application is launched, the user arrives at the login page. It is so possible to assign or un-assign a user from the web-console. Assign a user to a device brings the same functionality for auto logging as for personal device.

6.3 Application Management

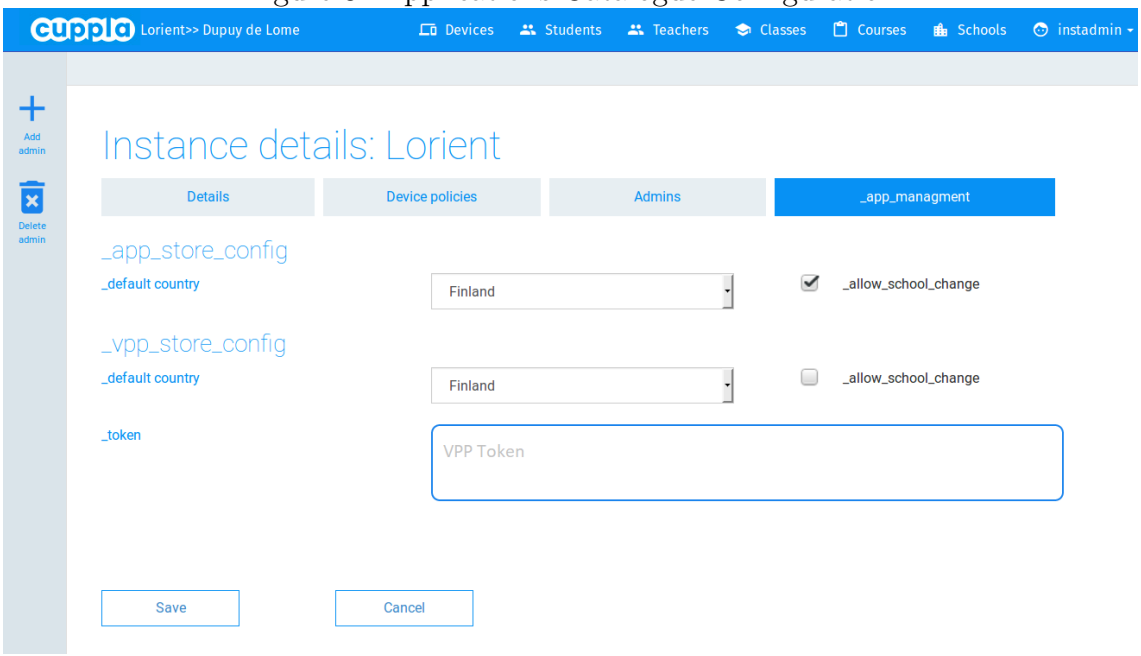
This section will detail the application management. It concerns the installation, un-installation, collect information about installed applications on devices, and search and add an application to the catalogue. Application management is another major feature for an MDM that is required for a school. Indeed, it simplifies the maintenance of a fleet of devices. For example, with only a few steps, it is possible to install or update an application for hundreds of devices.

Moreover, it ensures every device to have the minimum requirement of applications installed needed for a course. Indeed, mobile applications are part of the main utilisation of mobile devices inside classrooms. In this first part, the detailed process management of applications is described.

Applications which are added to the catalogue are divided into two categories. It is still possible to change an application to the other group. Applications are divided into the mandatory application and the optional apps.

Mandatory applications, as the name implies, are applications that have to be

Figure 3: Applications Catalogue Configuration



installed on every device. The first impact it has is during the enrollment phase. When the enrollment is finished, the device starts the check-in phase by sending information detailed, the server push installations of the applications marked as mandatory. However, it is possible for the owner of the device to manually uninstalled these applications from their devices, unintentionally, if they forgot it was a mandatory app. This is why it requires a mechanism to ensure they remain installed. For this, when the server updates the information of a device, it also queries the list of installed applications. Then, this list is compared with the mandatory applications, and if an application is missing, then a new installation command for this app is created.

Concerning optional apps, their installation is manual and can be operated from the action control in the web-console, or from the mobile client in the application catalogue tab, and manually install an application by selecting it.

The registered application in the catalogue is defined by a status. They can be either published or unpublished. Only published apps appear in the app list. Indeed it is not possible to install an unpublished application. A new app has by default the status published. To remove an application from the catalogue, it first requires to unpublished the selected app. There are two possible choices for unpublished an application; it is possible to uninstall this application from all the devices where it was installed by the MDM solution. If the app is not uninstalled, it will stay managed by the MDM server. That means it will be automatically uninstalled when the device will be unenrolled.

6.3.1 Store App

To be able to install an application to devices, it first requires to add them to the catalogue. The first kind of app that can be added is store applications. It concerns applications that are available in the different official stores (e.g., iTunes, Play Store).

The information needed to install an application from the store is its store ID and its identifier. If we take the example of an iOS app, the store ID can be found in the URL when opening the detail page of an app in iTunes. This is difficult and not convenient for the end-user to provide this ID when he wants to install an application. This is why to improve the ease of use, it is crucial to provide a search functionality which will gather all needed information and store them in the catalogue. The search tool allows the user to browse the different app from the different official stores. First, the user has to choose on which platform he desires to find the app (Apple or Android). Then, he can select from which country the server will get the information. Indeed, some application may not be available from all the country store. This limitation can come from legal reason, but mostly for language support. The possibility to select the country allows a better adaptation to international customers. Finally, the last parameter to launch a research is a selection of terms or keywords to filter the name of the application.

The research to the different stores are done with the URL used when browsing the actual stores. Unfortunately, there is no official APIs design to do a search. Therefore, REST communication is used to get a list of applications that match the previous criteria. If the user picks one app from the list, it will automatically fill in the form to add the application. The user can customise the entry by changing the name or the icon which will be displayed in the catalogue.

As mention previously, it is possible to select the country where the information is searched. To avoid the misuse of this feature by normal users, it is possible for the admins to configure the catalogue. The Figure 3 illustrates the catalogue configuration page at the instance level. Administrators can choose the default country which will be displayed first in the choice. It is also possible to forbid the selection of others supported countries, at the instance level to the school level.

The behaviour when installing an iOS or an Android application is different from the action that the end-user needs to perform. For both platform, when the device receives the command, it shows a pop-up message to the end-user. This user is asked to authorise the MDM server to install a new application on the device. In the case of iOS, accepting the message is enough to install the application. However, concerning the Android platform, the end-user is redirected to the Play Store. There, he will have to install the application normally.

Applications that are installed through the MDM solution are managed by the server; it is so possible to remove them from the devices. If an application were already installed on the device, the server would not push a new installation for this application. Therefore, applications installed by the users are not managed to preserve control of the users on their own device. This is important when using a BYOD policy to have a good equilibrium on the management of devices between owners and the MDM solution.

However, a simple way to restrict the utilisation of some applications for School Own Devices is to disable the public store. Therefore, only the MDM server will be able to send installation commands, and the choice of application will be limited to the custom store of the school.

6.3.2 Business App

For Android platform, it is possible to add customised applications. It can be internal applications (i.e., developed for the school) or B2B applications. The form to add a new application to the catalogue is slightly different from the one for store apps.

For this case, the user will have to enter the different information manually, like the name or the icon. There are two ways to the user to provide the application. The first method consists of uploading the app to the server. The APK file is stored in the database with the app entry. The second method to provide the application is to give the URL of the file. The URL can link to any location. Only the protocol HTTP is supported by the system. The link can be secured (HTTPS) or not.

For both cases, the installation process remains the same. The client sends the installation command and will first download the application on the local storage. After the download is finished, the user will be prompted to install the application as a tier app. The URL used to download the application, which is stored in the database, corresponds to the direct link to the file. That is why it has a similar behaviour as external applications. However, it can have some advantages to share only the URL of the file when adding the app to the catalogue. It is possible for the end-user to be located in a place with a bad network. Despite the slow connection, the internal network can still have correct performances. Therefore, it will take less time to download the file from a local server, compare to a remote cloud server.

Downloading the application on the local storage could seem a bad idea for the protection of the APK file at first glance. However, on Android, there is no concrete method to protect the access to these files to the end-users. A basic solution would be to forbid rooted device to use the MDM solution. In fact, the rooted devices might alter the execution of commands, or change system files. This is why a good practice is not to allow managing these devices. However, it does not solve the protection of APK files locally stored. When installing a new application, the system store the file, in order later on to execute the application. Moreover, some manufacturers which provide backup software are adding these files when the user wants to save his device. This explains why no concrete measure is taken, for now, to prevent user getting the APK when installing a custom app on their device. To prevent an external user from installing and using the application, the best solution would be to required an online authentication.

6.3.3 Volume Purchase Program App

The last kind of applications that are managed by the system concerns iOS applications. The Volume Purchase Program (VPP) from Apple, allows enterprise or schools to buy and distribute to their users, licences for applications. The applications which are compatible with this program are the same that are available on iTunes.

Description

This similarity of apps stores makes it possible to use the same installation process as for normal store apps. The only nuance is to add an option to specify the purchase method is using VPP. This flag means the device will check if a licence is assigned from VPP instead of from the registered user account.

When a user wants to add a VPP application, there are two possibilities. The first thing that needs to be done is to configure the VPP catalogue from the settings, as shown in Figue 3. The user has to specify his token which can be downloaded from his VPP account. This token is used to identify the account when using the Apple API.

As the application is similar to normal store app, the same searching tool, from the user view, is used for VPP applications. The REST request to the Apple server is made to the education VPP store, using the same information of country and keywords as for the public store research. If the user selects the iOS platform, he can enable or not the search for the VPP store. The adding process for the catalogue is similar than other applications. The only differences come from the fact, a new entry is added for the licence management.

Then, when the user goes on the detail page of this application, he can find a link which redirects him to the VPP store page of this application. From this page, he can buy licences for this applications, by connecting to his VPP account. Apple provides no options to purchase licences outside the VPP website. After having bought new licences, he needs to update the assets to update the number of licences. The total number of licences bought and the number of licences already used are shown in the application details.

This flow is the first method to add a new VPP application to the catalogue. The second method is when the user updates the assets from the token. If licences of application were already bought, and not present in the catalogue, a new entry is automatically created in the database. This is, for example, the case for custom business applications, or applications that are not available anymore in the store and so can not be found when using the search tool. When an application is not available, it is not possible to buy more licences. However, it is still possible to use the ones already bought.

Licence Type

There are three different ways to assign licences. The first one is to download a list of redemption codes. The two others are for managed distribution.

Redemption Code

If this solution is chosen, it is not possible to get back the licences bought, unlike the two others methods. The user has to download a file which contains the list of the redemption codes. The main problems of this method are to distribute the codes to users. It requires to send the code by email, and enter them manually for every user. Moreover, it is not possible to get back the licences. That means that when a user is adding the code to his account, it is like he bought the app himself. Thus, this solution

is not convenient for schools, where licences need to be re-assign every year or semester.

The two others methods allow to re-assign licences. These two approaches require having a Device Management solution to manage the licences. It is possible to assign a licence per user, and since the version 9.3 of iOS, to assign a licence to a device.

User Based

The advantage of user-based licence is that every application of the VPP store is compatible with this solution. Indeed, some applications are not compatible with the device-assignable feature. This limitation may be if some functionalities of the application require having an Apple ID.

When assigning a licence to a user, it binds the licence to an Apple ID. The major advantage is if the user wants to use the application on multiple devices (e.g., for a smartphone and a tablet). Thus, it reduces the number of licences needed to buy.

To assign a licence to a user, the first step to do is to invite the user to join the VPP account. It exists several possibilities to send a new invitation. The first one is sending the invitation by email. It is useful if the user has not enrolled a device yet. However, it can take time from the moment the invitation is sent and the moment the user accepts the invitation.

The second method is to use one device that the user already enrolled. For this case, a notification is sent to the device. A pop-up window asks the user to accept to join the VPP account link to the token. The user is redirected to the iTunes application to finalise the registration and links his account.

After a user account is registered to the VPP account, it is possible to assign licences. Then, to assign a licence, it requires to the user the same user ID which was used to register the user. As soon as the Apple server responds the assignation succeed, it is possible to send the installation command to a device, where the end-user sign-in with his registered account.

However, this solution is harder to use in practice with School Own Device. An account is limited in the number of installation with the same licence. Moreover, it requires having one account per device, which is not compatible with the solution used by some schools.

Device Based

The device-based licence was preferred compare to user-based licence, to support more use-cases. Therefore, after testing the user based solution and its limitation for School Own Device, device based method was included in the implementation. Moreover, only a few applications are not device-assignable. It is so compatible with the majority of use-cases of schools. With the device based licence, it is also possible to support School Own Devices which do not have Apple ID.

Unlike assigning a licence to a user account, for devices, there is no need to register first the device to the program. When assigning the licence, it just requires providing the serial number of the device so that it can be identified during the installation. The serial number is part of the information that is collected when the

device check-in after its enrollment. Therefore, there is no extra step to do for the assignation.

When the application is installed using the VPP flag, the device communicates with the Apple server to verify if a licence was already assigned. This method is independent of the user. The management of the licences in the system is explained in the following section.

Assignation

In the case of VPP applications, the assignation process is the first step to install applications. This is also required for free applications purchase with VPP. Indeed, paid or free VPP applications require a licence to be installed.

When an application is installed (requested by a user, or by the server for mandatory applications), before the install command is created, the server first initiate the assignation. The server communicates with the API by providing the token for identifying the VPP account, the serial number of device and the store ID of the application. Two kinds of response are considered as positive. The first one is if the status equal to '0', which means the association succeed, and the installation process can continue. The second case corresponds to the error code '9616', which signifies a licence is already assigned to this device for this application. All others error codes lead to aborting the installation process.

Local entries are used to keep a trace which devices hold a licence. They are used later on to un-assign licences. A licence is revoked from a device when the server orders the uninstallation of an application. That means licence are freed when an application is unpublished and uninstalled, or when a device is unenrolled.

One major drawback of the assignation comes from the token. It is used to authenticate the VPP account of the client. However, this token can be used in many different solutions at the same time. It does not exist a method to protect and block the utilisation of licences of one account to only one MDM solution. As no claim is possible, different MDM solutions can revoke the licences of a device, even if the licences were not assigned by this solution.

Supervised Device

As previously explained, devices can be supervised if they are managed by an MDM solution, or by using the Apple Configurator. When coupled with VPP, it presents some advantages.

The first one concerns the device assignation. Supervised devices belong to the schools, meaning they do not necessarily have Apple ID link to a device. An MDM that support VPP applications can overcome the problem of installing new apps, even with no iTunes account registered on the device. Unlike, Apple Configurator, there is no need to connect each device one by one. With the MDM, it is possible to install an application to all devices at the same time.

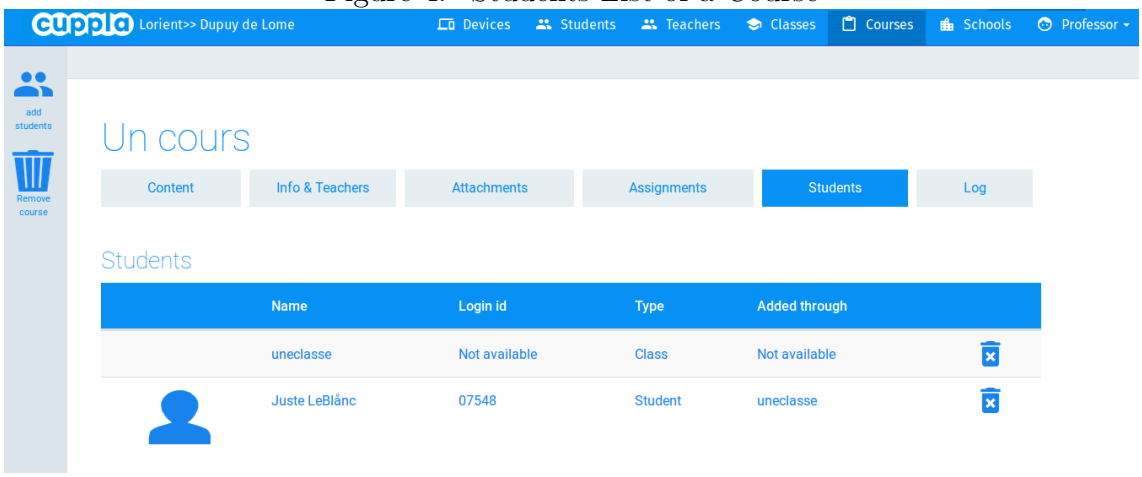
This is possible because of the second advantage of using supervised devices. When a device is supervised, the end-user does not have to accept the installation of

the application. The installation process becomes silent on the device, triggering no information to the end-user.

6.4 Course Management

The part will describe the mobile learning application and its advantages of using Device Management features. It will highlight the simplification of the utilisation, to enhance its accessibility.

Figure 4: Students List of a Course



6.4.1 Content Management

The application is focusing on course content sharing. This part will highlight the key points of the course management. The course management, as the others feature, is designed to reduce the complexity. When creating a course, the main teacher sets a beginning and ending dates to delimit the active period, when the course is active. He can share his course with other teachers.

Then, he has to register students. To face the different possibilities, it exists several ways to add students to a course. They can be added with their class. In that case, the class appears in the list of students as well as all the students. In the Figure 4, the student was included in the course with his class (in which he is the only member). It is also possible to add an individual student to a course. A teacher could remove students the same way if they were added individually or with their class.

When a student is added to a course, he gets access to it in the mobile application. An empty assignment is also created for him.

Content

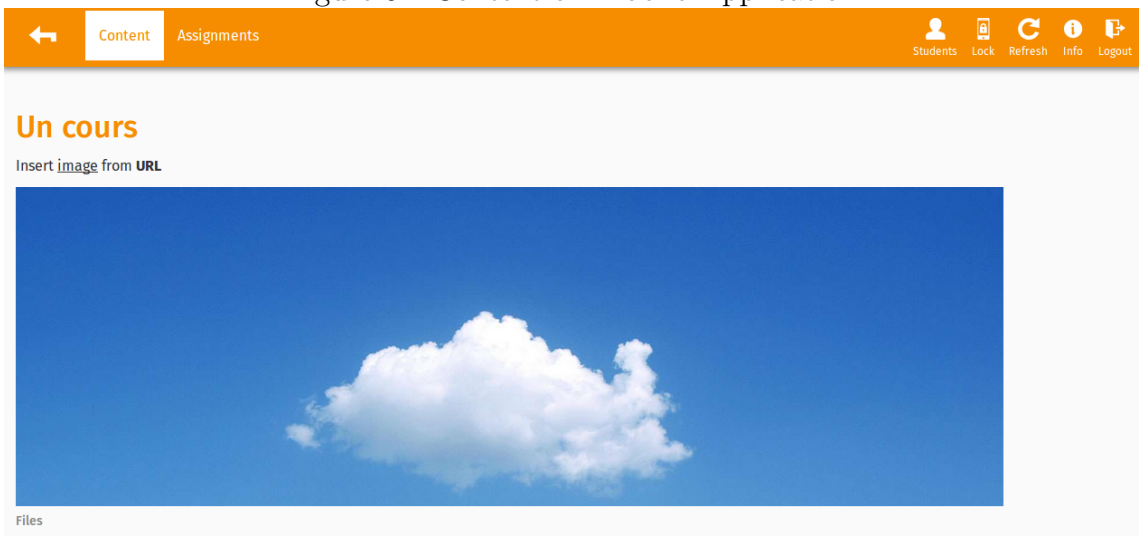
The course content is based on HTML. This support increases the compatibility with another system, compared to a built-in system. Thus, the integration of media

content is facilitated, allowing a freedom of creation. Among this media content, it is possible to embed images, as well as video.

The compatibility is also used to display this HTML content, with the web-view pages of the application. The solution is not platform dependent, and so can be easily integrated everywhere. As shown in the Figure 5, HTML content allows to include external content, like a picture, directly from its URL.

An important feature for the course content creation is the possibility to edit it quickly. Using a cross-platform solution allows to use content found elsewhere easily, but also permit to share templates between teachers.

Figure 5: Content on Mobile Application



Integration of Services

Another advantage of using an online solution is the integration of other solutions. There is no need to recreate already existing products. However, tiers application may require authentication from the users. Therefore, it may not be convenient for them to manage different sessions at the same time.

This is why the back-end server can be used to authenticate to other services. In that case, it just requires to setup the account initially at the instance or school level, like it is the case for VPP. Then the new service can be integrated with the initial solution and directly include inside courses.

6.4.2 Interaction

The interaction corresponds to the feature which allows the students to be active during a course. This solution currently only implements an assignment system.

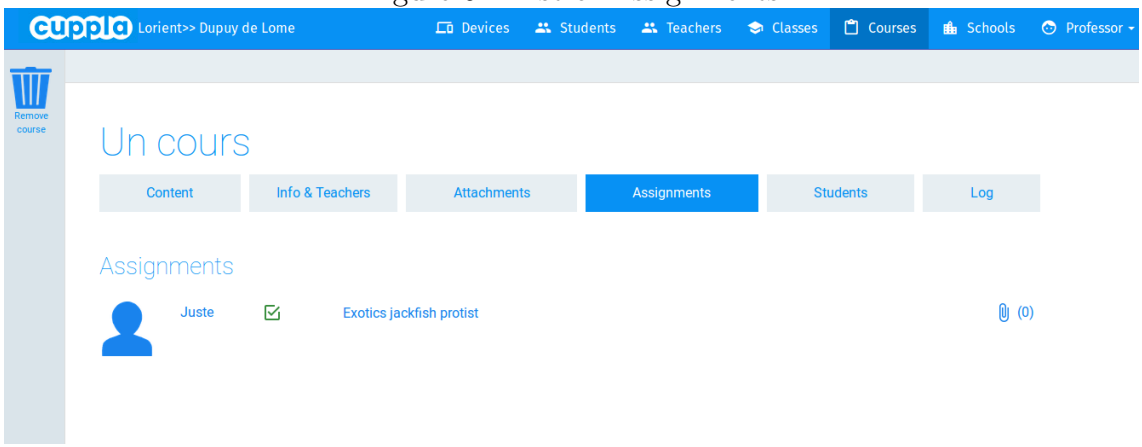
For mobile learning, the interaction between students and teachers is important to develop. The assignment is a way for a student to send for each course a text and some attachment files. It may be used for different purposes, such as sending

group work, doing homework. Moreover, it makes the possibility to have students more active using the application, instead of just passively watching and reading the content.

It is possible to add deadlines as assignments are editable only when the course is open, during its active period. After that time, students can only consult what they have produced.

As the rest of the content, the assignments are also directly saved online. It means if a student changes his device, he will continue to have access to his work. This is the case if students are using School Own Devices, they may not use the same device each time.

Figure 6: List of Assignments



Assignments are accessible for a teacher from either the application or the web-console. The list of the assignment of the student of the course is illustrated by the Figure 6. Then, by clicking on an item in the list, the teacher get the full text as well as attached files. From the mobile application, they have a quick preview of what the students have done. They can quickly get the result of the work of their students, and check their progress in real time.

If a student is removed from a course, his assignment is not deleted. In that case, it is only disabled and so disappears from all the views. This process is to avoid, errors from teachers (e.g., deleting a whole class instead of one student) and lose all the work of students. If the student is added back to the course, then his assignment is restored. Assignments live as long as the course where they belong.

7 Evaluation

This section will evaluate the implementation of the solution described in the previous section. As it is not based on performances but on features, no benchmark is used to analyse this solution. Instead, the features will be evaluate based on the problem statement proposed in the Section 3.2. Moreover, these features will also be compared with those provided by MDM competitors used for education.

The different competitors will be briefly presented at the end of this section, in the Section 7.5.

7.1 User Management

As described in the problem statement, the system have to propose a single user based. This user system should give access to all the different services. Moreover, to facilitate the usage of the system, it might divide the users in specific roles.

7.1.1 Unique System Infrastructure

The unique system infrastructure is based on two principle. The first one is the division of the system for customers. The second uniqueness is to regroup both device and learning management in one place.

Structure of the System
For this implementation, the customers own one instance in the system. It can regroup one or many schools, depending if the customer is a school or a group of schools (e.g. a city).

Unlike others MDM solution which are specialised for schools, it is possible to share a same service between several schools that belong to the same groups. For instance, it allows to share a VPP account at the instance level and so avoid licence inconstancies.

All-in-one
This implementation provides both device and learning management. Unlike MDM solutions that were first developed for enterprise, schools do not require to use another system for their courses. The solution previously presented manage all different features from only one user base.

Some MDM solutions are directly specialised for schools. They are based on Apple products and offer a full compatibility with the Apple services. So that they are using Apple School Manager (ASM), in order to manage the class level. They ensure to use only one system by importing students from the Apple classroom directly to their system. However, even if they have a common entry for users, these solutions still used two user experiences for devices and courses management.

7.1.2 User Roles

The main advantage of the solution is that it was directly designed for education. Thus, it is developed to face the requirements and concerns of schools. The main

end-users are teachers and students, that means users who may not used to this kind of technology. Therefore, the division of tasks between separate actors is important to ensure that the solution is accessible to a maximum of end-users. It is possible to have an IT user only for the instance, and easily manage multiple schools. Because of pre-configuration and inheritance of parameters, the solution is ready out of the box. Then, teachers can only focus on the course management, letting the system taking care of the rest.

Another MDM provider go further in the possible roles. ZuluDesk also provides a role for parents in order to look after the student activity even after school time, and ensure a continuous utilisation from school to home.

7.1.3 Interfaces

As this solution is developed for school, its interface is adapted to the end-users. Indeed, they are not expected to be experimented like the IT department, and so requires an easier interface. Therefore, the user interfaces are light and easy to use. The end-user has to deal with only the minimum information required. The different functionalities are well divided, in order to keep a clear view for each page. Moreover, every configuration requires only few step, before completing the action.

It is the case for adding a new application, where only the name of the application is required, the server automatically fills the other fields. If it set as a mandatory application, it will be automatically installed the next time devices will update connect to the server to their information.

Another example is when a student is added to a new class. It can be the case, when a new student is added to the system, he needs to belong to one class. It can also be the case if he is transferred from one class to another. Then, the system automatically removed him from his previous courses, and add him to the courses link to that class. If he has common courses, he will, of course, keep the same assignment, as they are only disabled when leaving a course.

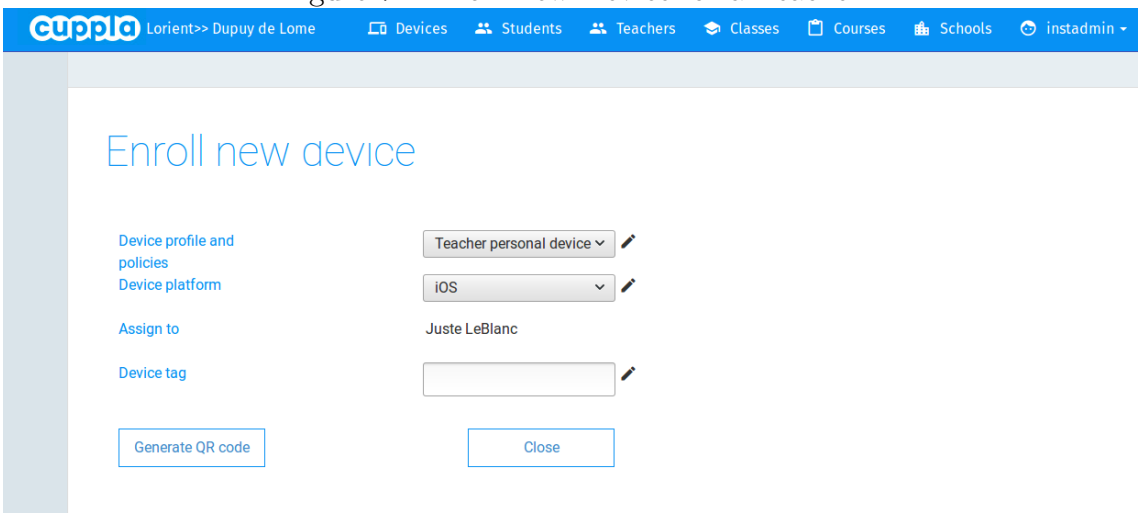
7.2 Device Management

The device maintenance might ensure the registration and the maintenance of devices in the system. It also needs to provide services to control devices when used in the classrooms.

7.2.1 Enrolling

Like most of the others MDM provider, the enrolling process tends to be as quick and easy as possible. This is why the server directly manages all technical information such as the address of the server during the token creation. Moreover, the information asked for the token is reduced when enrolling a device from a specific user page. In that case, only the platform of the device is required, as shown in the Figure 7. As even this information is getting during the device check-in, this information will no longer be required to generate a new token. The server automatically initializes the others value, such as user, instance, and school names.

Figure 7: Enroll New Device for a Teacher



Heterogeneity
One major advantage concerning the Device Management part is the cross-platform ability. It allows a heterogeneity of devices facing all possible solutions. As it can be difficult to learn many things simultaneously, it is important for the user to adopt a new product by using their own mobile device. Thus, the multi-compatibility become a key point for such a solution.

For the competitors, those who are compatible with learning management use only the Apple product and so are not cross platform.

7.2.2 Command

The command section follows the specifications for Device Management. From a user point of view, on the interface of the web-console, one command will have the same process independently of the platform. This permits to support any devices and provide the same user experience even for BYOD devices.

The Device Management features also replaced solution like Apple Configurator. It provides similar configurations for devices, sending them over the air. However, unlike the Apple software, there is no need to configure devices one by one, by connecting them to the configuration machine. The MDM server automates all the actions by pushing modifications to devices. Every action initiated by an end-user takes up from the cloud and then can be done everywhere. Moreover, services offer to users require a few information and step from the homepage to action control page.

7.2.3 Services

Pre-defined values for policies are pre-registered by default. These values should fit for most of the customers. Thus, the product is ready to be used out of the box.

Network
The main limitation of this solution is the network dependencies. The course management is based on web services. It ensures to always have the last version of a course or an assignment. It also permits users to use different devices each time, in the case of School Own Devices, and always have access to their own session. However, it becomes impossible to use devices offline to access either the content or the management services. Examples of school trips outside (e.g., museum visits), where no WI-FI access is provided.

Some schools may not have a network speed efficient enough for managing devices. This is partly solved for the installation of applications by using custom URL for applications stored on a local server. It is also possible to store media content of a course locally to reduce the required bandwidth.

The Device Management is also possible only when the device is connected to the internet. It may create some rush hours after weekends or holidays. Many devices will not connect during a long lapse of time, and exceed the delta time for updating their information on the server. In the case no devices connect during the regular update time, they will all need to update at the same time. Therefore, it will require waiting that all bunch of devices are selected during the routines. Thus, to update all the devices it needs to wait the delta time fixed for each instance hence may require several hours to update a device.

7.3 Application Management

The problem statement specifies that the system must provide a complete application management. The solution needs to take into account a large variety of possible cases of installation processes and kinds of applications. The same interface is used for the user to manage applications, for Android and Apple platform. Applications can be installed from the web interface or directly from the device itself using the mobile client.

The solution that is implemented for the application management depends on the API and implementation of each platform. Therefore, it is quite similar to what others competitors can offer. Indeed, this feature is part of the core functionality of MDM. For instance, it is quite common to support custom applications and personalised applications catalogue for enterprise.

App type
The application management, like the others part of the solution, is implemented to facilitate its utilisation by any one. This start with the search tool to add a new app to the catalogue of a school which requires only the name of the app and its platform. Then, when browsing the device detail, only the list of compatible applications for this device is displayed, and indicate the installation status.

During the installation process, the current status of installation is indicated, i.e., installed or pending, or not installed. When selecting an application, the server automatically chooses which kind of installation is required depending on the type of apps. The system differentiates app that needs to be installed from the public store,

the ones using VPP or the files stored in the cloud, or accessible from an external URL.

App Data
In the case an application is unpublished and uninstalled, the server sends uninstallation commands to devices. However, if the application is, later on, reinstall to devices, the system will not restore previous data of this application. There is no back-up done except if taking cared by the application itself or another service linked to the account. Unlike other MDM solution developed for enterprises, there is no management of data locally stored on devices. No security system was implemented to protect data except the factory reset.

7.4 Course Management

As the solution focuses on mobile learning, the system developed have to include a course sharing tool. This tool is used to provide content as well as to improve the interaction inside and outside classrooms.

7.4.1 Content Management

This solution does not have real competitor among MDM solution providers. Some of the competitors who are focusing on schools as customers are compatible with course management solutions. They are only compatible with Apple devices; therefore, they can provide an integration of Apple services. These services will bring all the course layer to their own solution.

Course Content
The course management is focused as a creative tool. It allows lots of freedom for teachers in selected their content, and which media to use. However, currently, no templates are provided to guide the ones with fewer abilities using computers. It also requires a transition to recreate already existing course on legacy support to this solution.

7.4.2 Interaction

The interaction is the missing part of this implementation compare to what the problem statement planned. In this solution, it is quite limited and only implemented with the assignments. These are limited to one per student per course and is designed only for individual work.

With the current implementation, it is not possible to organise group works and develop interaction between students. Moreover, after the class, there is no communication means with the teachers to bring the course to a level outside the classroom. Further possible implementations are added in the future work part.

Table 1: Competitor Analysis

	Single System	Services for Classroom	BYOD	Cross Platform	Business Apps	VPP Apps
Implementation	V	V	V	V	V	V
Mobile Iron	X	X	V	V	V	V
AirWatch	X	X	V	V	V	V
Cisco Meraki	X	X	V	V	X	V
Miradore Online	X	V	V	V	V	V
Hexnode	X	V	V	V	V	V
Lightspeed	X	V	V	V	V	V
Zuludesk	ASM	V	X	X	X	V
Mosyle	ASM	V	V	X	X	V
Jamf	ASM	V	V	X	V	V

7.5 Competitors Analysis

The MDM providers that compete with the presented solution are divided into two groups. The first one concerned the solution compatible with multiple mobile platforms. The second group focus only on one Operating System, iOS for all of them.

The features they are providing are shown in the Table 1. The features consist of the following:

- Single System: device and course management.
- Services for Classroom: controls devices in classroom (e.g., lock, restrict content).
- BYOD: supports Bring Your Own Device policy.
- Cross Platform: supports all major mobile OS.
- Business App: allows installation of custom apps.
- VPP App: compatible with Apple VPP service.

Multi-platform

The main MDM solution that are compatible with multiple mobile OS are: MobileIron[9], Airwatch[10], Cisco Meraki[11], Miradore Online[12], Hexnode[13] and Lightspeed[14]. The three firsts were initially developed for enterprises. However, thanks to their popularity, they are adopted by some schools that needed an MDM solution. Therefore, they have more advanced features on data management and security of devices and information.

[9] https://www.mobileiron.com/en/solutions/mobile-device-management-mdm
[10] https://air-watch.com/en/solutions/
[11] https://meraki.cisco.com/products/systems-manager
[12] https://www.miradore.com/mobile-device-management-for-education/
[13] https://www.hexnode.com/mobile-device-management/
[14] http://www.lightspeedsystems.com/products/mobile-manager/

The last three of the list are specialised for schools. Even if they do not provide a course management system, they introduce some functionalities to facilitate the utilisation of mobile devices inside classes. Thus, they provide the core MDM services (e.g., internal app catalogue), plus device control inside the classroom, such as, lock, or sending books for the course content.

Apple Device Solutions

The second category of MDM solution regroups providers specialised for mobile learning: Zuludesk[15], Mosyle[16], and Jamf[17]. They all based their solution on the compatibility with Apple classroom services. Therefore, they only allow the utilisation of Apple devices with their product. Then, they can ensure better control using supervised devices, by supporting the DEP Apple program.

To provide only one common system with the Apple course management, they use the ASM. Therefore, they can import users and the class level of their solution. To provide a full solution and improve the device usage, Zuludesk also includes parents as an actor in the system.

7.6 Future Work

This part aims to propose some ideas to improve the overall user experience. It concerns mostly features that may directly impact the end-users. These new possible implementations will bring more value for customers, compare to an improvement of performances for the system. Indeed, such a product is more focus on users needs and customers interests.

Screen Sharing

The major addition that can be done is screen monitor in the mobile application. Inside a course, a teacher would be able to have access to the display of students who entered in the same course. The idea share similarities with the AirPlay feature developed by Apple which can be integrated into iOS product with the MDM server. On the contrary of the Apple functionality, it may stay cross platform as the others feature already provided by the solution.

This screen monitor can be used for two purposes. It may be used to look after student activity when they are using their mobile devices. Therefore, it will ensure the possible distraction from students during some activities. The second possibility would be the screen sharing. For that case, it may be used to cast the screen of a particular device (teacher or student) to all others devices, to highlight one specific work.

This kind of feature will, of course, require additional network implementation to control the required bandwidth and the latency.

[15] https://zuludesk.com/features/
[16] http://manager.mosyle.com/
[17] https://www.jamf.com/solutions/technologies/classroom-management/

Course Creation

As previously explained, the course content can include multiple kinds of media (e.g., text, pictures, video) and let lots of freedom of creation. However, this tool may stay difficult to apprehend for a reluctant teacher to new technologies. It may also be difficult to create a course that looks appealing.

Assisted tool creation for course content can solve some issues creation. It can be base on templates pre-created or shared on a platform. A block based creation tool can also be considered. It would help the creation of the content by adding blocks automatically generated based on the content provided as parameter, similarly as a pretty printer.

Course Analytics

A key point for a course is the feedback a teacher received. It can take the form of work, like for example with the assignment.

However, it can also be with analytics on different criteria. These criteria can be on time spent by students on each part of a course. Therefore, it will give an indication to the teacher which parts are more challenging and require more explanations.

For instance, it also can measure the global time spent in a course. It may give insight into the importance and the impact of using mobile devices for courses.

Interaction

The interaction may be improved on two different points. The first aspect is to continue the development of assignment by adding new features. It can be improved with more types of assignment, such as group work, integration of quiz, and the possibility to return several assignments per student.

The other way to add more interaction would be with communication tools. It can help to develop activities outside the classroom, by keeping a link between students and also with their professors. Unlike traditional communication tools, students may be able to add comments along the course content to ask questions for specific parts.

8 Summary

With the digitalization of our society, it becomes necessary to learn digital skills at schools. One aspect of mobile learning is to use this new technology directly during classes to introduce new competencies. Moreover, using mobile devices can extend courses to more content, including more interaction. This interaction comes from either of the content itself, for example with the application, or thanks to the mobility provided by the device, and hence change the environment.

Therefore, it changes the traditional way of teaching to adapt to modern concerns. This result that students are more involved in their studies with more informal methods. However, it gets a better result when there is one device per student during a course. So, even if many solutions were already developed, they suffer from major lacks to be self-sufficient. The main required feature is to be able to manage all these devices, even personal ones.

Mobile learning based on Device Management tends to remove the main issue when using the device: setup and maintenance. The solution presented in this thesis shows a possible integration of MDM developed specifically for education. This result by taking care of the main requirements of schools that may be different from the MDM developed for enterprises. Thus, it will focus less on security and privacy of shared data (e.g., course contents). Hence, MDM server still provides main security features, such as device authentication to access cloud services.

Indeed, thanks to an MDM solution, schools are ensured that students and teachers will use devices which are correctly setup. It also brings additional features that improve the mobile learning solution. The possibility to lock the devices of students during a course may reduce the possibility of distraction during classes. Therefore, it ensures students to use their devices efficiently during class time.

The device setup and maintenance also include application management. This represents an important feature for schools, as devices are mainly used for applications. Therefore, it is possible for a school to allow BYOD policy while managing all content for every kind of devices.

This MDM solution represents a unique central system used to take care of main features. Thus, additional web services can be added to improve the whole solution. These external services can be used to fill some lack of the course system, such as adding more interaction during courses. As the content creation is HTML base, it is so possible to integrate other product such as Kahoot! to enhance the global experience of students.

The evolution of learning process will tend to apply the principle of Mobile Learning, thus, increase the global utilisation of mobile devices inside classrooms. This evolution is following the digitalisation of content, from traditional books to e-books and mobile applications.

References

[1] Rahat Afreen. Bring your own device (BYOD) in higher education: opportunities and challenges. *International Journal of Emerging Trends & Technology in Computer Science*, 3(1):233–236, 2014.

[2] Open Mobile Alliance. Oma Device Management Bootstrap. *Candidate Version*, 1:1–35, 2012.

[3] Open Mobile Alliance. Oma Device Management Protocol. *Approved Version*, 1(1):60, 2016.

[4] Open Mobile Alliance. Oma Device Management Security. *Approved Version*, 1, 2016.

[5] Rashmi A Bajad, Monika Srivastava, and Amit Sinha. Survey on mobile cloud computing. *International Journal of Engineering Sciences & Emerging Technologies*, 1(2):8–19, 2012.

[6] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment*, 8, 2013.

[7] Tak-Wai Chan, Jeremy Roschelle, Sherry Hsi, Kinshuk, Mike Sharples, Tom Brown, Charles Patton, John Cherniavsky, ROY PEA, Cathie Norris, et al. One-to-one technology-enhanced learning: An opportunity for global research collaboration. *Research and Practice in Technology Enhanced Learning*, 1(01):3–29, 2006.

[8] Soumya Kanti Datta and Christian Bonnet. A lightweight framework for efficient M2M device management in oneM2M architecture. In *Recent Advances in Internet of Things (RIoT), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[9] Ryan Dellos. Kahoot! A digital game resource for learning. *INSTRUCTIONAL TECHNOLOGY*, 49, 2015.

[10] Inge deWaard, Apostolos Koutropoulos, N Keskin, Sean C Abajian, Rebecca Hogue, C Osvaldo Rodriguez, and Michael Sean Gallagher. Exploring the MOOC format as a pedagogical approach for mlearning. In *Proceedings of 10th World Conference on Mobile and Contextual Learning*, pages 138–145, 2011.

[11] Mohamed Osman M El-Hussein and Johannes C Cronje. Defining mobile learning in the higher education landscape. *Journal of Educational Technology & Society*, 13(3):12, 2010.

[12] Beth Fuchs. The writing is on the wall: Using padlet for whole-class engagement. *LOEX Quarterly*, 40(4):7, 2014.

[13] Ignacio Garcia. Learning a language for free while translating the Web. Does duolingo work? *International Journal of English Linguistics*, 3(1):19, 2013.

[14] Arnab Ghosh, Prashant Kumar Gajar, and Shashikant Rai. Bring your own device (BYOD): Security risks and mitigating strategies. *Journal of Global Research in Computer Science*, 4(4):62–70, 2013.

[15] Peter Goodyear and Symeon Retalis. Technology-enhanced learning. *Rotterdam: Sense Publishers*, 2010.

[16] Philip J Guo, Juho Kim, and Rob Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.

[17] Matthew Kearney, Sandra Schuck, Kevin Burden, and Peter Aubusson. Viewing mobile learning from a pedagogical perspective. *Research in learning technology*, 20, 2012.

[18] Hanan Khalil and Martin Ebner. MOOCs completion rates and possible methods to improve retention-a literature review. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, volume 2014, pages 1305–1313, 2014.

[19] Adrian Kirkwood and Linda Price. Technology Enhanced Learning and teaching in higher education: What is 'enhanced'and how do we know? A critical literature review. *Learning, media and technology*, 39(1):6–36, 2014.

[20] Agnes Kukulska-Hulme. Will mobile learning change language learning? *ReCALL*, 21(02):157–165, 2009.

[21] Teresa Martín-Blas and Ana Serrano-Fernández. The role of new technologies in the learning process: Moodle as a teaching tool in physics. *Computers & Education*, 52(1):35–44, 2009.

[22] Alexander McAuley, Bonnie Stewart, George Siemens, and Dave Cormier. The MOOC model for digital practice. 2010.

[23] Keith W Miller, Jeffrey Voas, and George F Hurlburt. BYOD: Security and privacy considerations. *It Professional*, 14(5):53–55, 2012.

[24] Daniel FO Onah, Jane Sinclair, and Russell Boyatt. Dropout rates of massive open online courses: Behavioural patterns. *EDULEARN14 Proceedings*, pages 5825–5834, 2014.

[25] Keunwoo Rhee, Woongryul Jeon, and Dongho Won. Security requirements of a mobile device management system. *International Journal of Security and Its Applications*, 6(2):353–358, 2012.

[26] Keunwoo Rhee, Dongho Won, Sang-Woon Jang, Sooyoung Chae, and Sangwoo Park. Threat modeling of a mobile device management system for secure smart work. *Electronic Commerce Research*, 13(3):243–256, 2013.

[27] Alexander Samochadin, Dmitry Raychuk, Nikita Voinov, Dmitry Ivanchenko, and Igor Khmelkov. MDM based mobile services in universities. *International Journal of Information Technology & Computer Science (IJITCS)*, 13(2):35–41, 2014.

[28] Mike Sharples, Josie Taylor, and Giasemi Vavoula. A theory of learning for the mobile age. In *Medienbildung in neuen Kulturräumen*, pages 87–99. Springer, 2010.

[29] Jane Sinclair, Russell Boyatt, Jonathan Foss, and Claire Rocks. A tale of two modes: Initial reflections on an innovative MOOC. In *International Workshop on Learning Technology for Education in Cloud*, pages 49–60. Springer, 2014.

[30] Giasemi Vavoula and Mike Sharples. Meeting the challenges in evaluating mobile learning: A 3-level evaluation framework. *International Journal of Mobile and Blended Learning*, 1:54–75, 2009.

[31] Giasemi Vavoula, Mike Sharples, Paul Rudman, Peter Lonsdale, and Julia Meek. Learning bridges: A role for mobile technologies in education. *Educational Technology*, 47:33–36, 2007.

[32] Roumen Vesselinov and John Grego. Duolingo effectiveness study. *City University of New York, USA*, 2012.

[33] Y Vovides, S Sanchez-Alonso, V Mitropoulou, and G Nickmans. The use of e-learning course management systems to support learning strategies and to improve self-regulated learning. *Educational Research Review*, 2(1):64–74, 2007.

[34] Darrell M West. Mobile learning: Transforming education, engaging students, and improving outcomes. *Brookings Policy Report*, 2013.

A VPP implementation

This code shows the communication with the VPP servers to assign a new licence to a device. To un-assign a licence, it follows the same process, changing the main command to disassociation.

```
#Assign a VPP licence to a device

def assignLicense(cls, device, app):
        try:
                #get the service url
                data = urllib2.urlopen('https://vpp.itunes.apple
                    .com/WebObjects/MZFinance.woa/wa/
                    VPPServiceConfigSrv').read()
                json_data = json.loads(data)
                url = json_data["manageVPPLicensesByAdamIdSrvUrl
                    "]
                #assign a licence to this device
                stoken = device.school.token
                query_args = { 'adamIdStr':app.store_id, '
                    princingParam':'STDQ', '
                    associateSerialNumbers':device.serial_number,
                     'sToken':stoken }
                context = urllib.urlencode(query_args)
                request = urllib2.Request(url, context)
                data = urllib2.urlopen(request).read()
                json_data = json.loads(data)
                if json_data['status'] == 0:#a licence was
                    correctly assigned
                  VPPDeviceAssigned.objects.create(device=device
                      , adamIdStr=app.store_id)
                  #update licence count
                  licence = VPPLicence.objects.get(school=device
                      .school, adamIdStr=app.store_id)
                  licence.assignedCount = licence.assignedCount
                      + 1
                  licence.save()
                elif json_data['associations'][0]['errorCode']
                    == 9616:#licence already assigned for this
                    device
                         VPPDeviceAssigned.objects.get_or_create(
                            device=device, adamIdStr=app.store_id
                            )
                #the licence is assign to the device
  except:
        #an error occured: not enough licence, or network issue
```

This code shows the communications to get the list of licences bought for VPP applications, and the process to add a new application.

```
#Update the licence count from the VPP account

def vppUpdateAssets(school):
        #get the service url
        data = urllib2.urlopen('https://vpp.itunes.apple.com/
           WebObjects/MZFinance.woa/wa/VPPServiceConfigSrv').
           read()
        json_data = json.loads(data)
        url = json_data["getVPPAssetsSrvUrl"]
        #get assets associate to this token
        stoken = school.token
        query_args = { 'includeLicenseCounts': 'true', 'sToken':
           stoken }
        context = urllib.urlencode(query_args)
        request = urllib2.Request(url, context)
        data = urllib2.urlopen(request).read()
        json_data = json.loads(data)
        #update licences
        if json_data['status'] == 0:
          if 'assets' in json_data:
                assets = json_data['assets']
                for asset in assets:
                VPPLicence.objects.update_or_create(school=
                   school, adamIdStr=asset['adamIdStr'],
                  defaults={'assignedCount':asset['assignedCount
                     '], 'totalCount':asset['totalCount']},
                  )
                q_instance = Q(instance=school.instance)
                q_no_school = Q(school=None)
                q_school = Q(school=school)
                q_os = Q(os=Enums.OS_IOS)
                if AppContent.objects.filter(((q_instance &
                   q_no_school) | q_school) & q_os & Q(store_id=
                   asset['adamIdStr'])).count() == 0: #create a
                   new entry if not added before
                        addApp(school, asset['adamIdStr'])

def addApp(school, app_id):
        stoken = school.token
        country = school.vpp_store_country #as define in the
           catalogue setting
        #get the application information from Apple services
        #the token is used in case the application is not
           available any more, or for business applications
        #it is still possible to get information for already
           bought licences
        STORE_URL = 'https://uclient-api.itunes.apple.com/
           WebObjects/MZStorePlatform.woa/wa/lookup?'
        STORE_ID = '&id='+str(app_id)
        END_URL = '&p=mdm-lockup&caller=MDM&platform=itunes&cc='
```

```
            +country
        header = urllib2.build_opener()
        header.addheaders.append(('Cookie', 'itvt='+stoken))
        data = header.open(STORE_URL + STORE_ID + END_URL).read
            ()
        json_data = json.loads(data)
        entry = json_data['results'][app_id]
        name = entry['name']
        instance = school.instance
        type_content = 'APP'
        os = Enums.OS_IOS
        identifier = str(entry['bundleId'])
        url = str(entry['url'])
        version = str(entry['offers'][0]['version']['display'])
        pic = str(entry['artwork'][0]['url'])
        AppContent.objects.create(name=name, school=school,
            instance=instance, type=type_content, os=os,
            identifier=identifier, store_id=app_id, url=url,
            version=version, thumbnail_url=pic,
            store_country_code=country)
```

B Device Update

This code illustrates the task executed by a routine to update the device details.

```
#Update device information

def update_device_info():
    #perform this task for each instance in the system
    for instance in Instance.objects.all():
        devices = Device.objects.filter(school__instance=
           instance, mdm_status=Device.ENROLLED)
        #get the delta time for this action
        try:
          time_info = Task.objects.filter(task=Task.DEVICE_INFO)
             .get(instance=instance).time
        except Task.DoesNotExist:
          time_info = 8
        nb_info = devices.count() / (time_info * 4) + 1

        for device in devices.order_by('last_device_info'):

            if device.last_device_info < timezone.now()-
               timedelta(seconds=3600*time_info):
                device.update_device_information()
                device.save()
                nb_info-=1
                if nb_info <= 0:
                    break
            else:
                break

        #get the delta time for this action
        try:
          time_app = Task.objects.filter(task=Task.APP_LIST).get
             (instance=instance).time
        except Task.DoesNotExist:
          time_app = 8
        nb_app = devices.count() / (time_app * 4) + 1

        for device in devices.order_by('last_app_list'):
            #uninstall application when they are 'unpublished
               and uninstalled'
            apps_needed_to_uninstall(device)

            if device.last_app_list < timezone.now()-timedelta(
               seconds=3600*time_app):
                #if an mandatory application is missing then an
                   installation command is sent
                device.update_applicationlist()
                device.save()
                nb_app-=1
                if nb_app <= 0:
                    break
```

```
            else:
                break


        #auto un-enroll devices after 90 days
        for device in devices.order_by('last_connected'):
          if device.last_sync_time < timezone.now()-timedelta(
              days=90):
            device.unenroll()
          else:
            break
```